CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

PHD THESIS

Deep learning techniques have been widely utilized across various domains including autonomous driving, content generation/recommendation, drug discovery, voice assistants, and renewable energy management. Ensuring the reliability of deployed models in the real-world applications has become more critical than ever. This thesis aims to enhance the reliability of deep models for trustworthy artificial intelligence by addressing out-of distribution detection (OOD), model calibration, and hallucination mitigation.

The key results in this thesis reveal the following insights: 1) Training deep models utilizing joint energy-based modeling enhances OOD detection performance and results in better calibrated regressors and classifiers. 2) OOD detection can be effectively achieved by utilizing only information available in the probability space of discriminative classifiers; 3) Medical anomalies can be identified using only normal images. By utilizing transfer learning and self-supervised learning techniques, an efficient feature-based framework is developed to detect medical anomalies in Chest X-rays. This approach outperforms reconstruction-based methods in terms of accuracy and effectiveness; 4) The knowledge of OOD detection within the framework of discriminative classifiers, can be effectively transferred to contrastive vision-language models (VLMs), enabling zero-shot OOD detection; 5) The insight gained from OOD detection has potential to address object hallucination in generative VLMs.

XIXI LIU • Towards Reliable Deep Foundation Models • 2025

Towards Reliable Deep Foundation Models

in OOD detection, model calibration, and hallucination mitigation

XIXI LIU

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Towards Reliable Deep Foundation Models

in OOD detection, model calibration, and hallucination mitigation

Xixi Liu

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden, 2025

Towards Reliable Deep Foundation Models

in OOD detection, model calibration, and hallucination mitigation

Xixi Liu
ISBN 978-91-8103-158-4

Doktorsavhandlingar vid Chalmers tekniska högskola
Ny serie 5616
ISSN 0346-718X

Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden
Phone: +46 (0)72 975 5242

Printed by Chalmers digitaltryck
Gothenburg, Sweden, March 2025

Knowing the path is different from walking the path.

Towards Reliable Deep Foundation Models

in OOD detection, model calibration, and hallucination mitigation

Xixi Liu
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Despite the success and potential of deep learning techniques, ensuring the reliable deployment of such models remains a primary concern. In this thesis, the reliability of deep models is tackled through the lens of out-of-distribution (OOD) detection, model calibration, and hallucination mitigation, contributing to a trustworthy artificial intelligence (AI) system.

Paper A and Paper B utilize joint energy-based modeling (JEM), and develop a probabilistic classifier and regressor, respectively. Specifically, Paper A addresses the training instability of joint energy-based models by replacing stochastic gradient Langevin dynamics with slice score matching, which results in a smoother training procedure without compromising the OOD performance. Paper B extends the idea of JEM from classification to regression, leading to a better calibrated regressor.

Paper C focuses on large-scale OOD detection with standard discriminative classifiers and proposes a novel OOD score based on generalized entropy, utilizing only information from the probability space.

Paper D leverages transfer learning and self-supervised learning techniques to devise an efficient framework, in which only normal samples are required for detecting anomalies in Chest X-rays.

Paper E utilizes the powerful text-image alignment in contrastive vision-language models (VLMs) for zero-shot OOD detection.

Finally, Paper F leverages insights from OOD detection and proposes an energy-based decoding method to mitigate object hallucination in generative VLMs.

Keywords: Trustworthy AI, VLMs, uncertainty estimation, OOD detection, model calibration, hallucination mitigation

List of Publications

This thesis is based on the following publications:

[A] **Xixi Liu**, D Staudt, Che-Tsung Lin, Christopher Zach, "Effortless Training of Joint Energy-Based Models with Sliced Score Matching". Published in International Conference on Pattern Recognition (ICPR), Montreal, Canada, August, 2022.

[B] **Xixi Liu**, Che-Tsung Lin, Christopher Zach, "Energy-based Models for Deep Probabilistic Regression". Published in International Conference on Pattern Recognition (ICPR), Montreal, Canada, August, 2022.

[C] **Xixi Liu**, Yaroslava Lochman, Christopher Zach, "GEN: Pushing the limits of softmax-based out-of-distribution detection". Published in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, Canada, June. 2023.

[D] **Xixi Liu**, Jennifer Alvén, Ida Häggström, Christopher Zach, "Deep Nearest Neighbors for Anomaly Detection in Chest X-Rays". Published in International Workshop on Machine Learning in Medical Imaging, held in conjunction with MICCAI, Vancouver, Canada, October. 2023.

[E] **Xixi Liu**, Christopher Zach, "TAG: Text Prompt Augmentation for Zero-Shot Out-of-Distribution Detection". Published in 18th European Conference on Computer Vision (ECCV), Milan, Italy, October. 2024.

[F] **Xixi Liu**, Ailin Deng, Christopher Zach, "Energy-Guided Decoding for Object Hallucination Mitigation". Submitted for Review, 2025.

Acknowledgments

First, I would like to thank my supervisor, Christopher Zach, for offering me this opportunity to start my Ph.D. journey during the Covid-19. Christopher told me once during the lunch time, " Let's see whether you can find your *calling* in the next five years.", which guides me through these year. I am sure I will keep asking myself this question in the future. I thank Christopher for allowing me to explore the topics I feel passionate about, and always reminding me that the goal of this journey is to be an independent researcher. I would like to take a moment to thank Ida Häggström for all the support and encouragement she has given me. Your generosity and belief in me have had a significant impact on my Ph.D. journey and will continue shaping my academic journey!

Further, I would like to thank my collaborators, Alex, Dorian, Yara, Ida, Jennifer, and Ailin for their great work! In particualar, I feel grateful for valuable research papers shared by Alex in weekly meetings, and engaging research discussions with Ailin. A big thank to Huu Le, Shuangshuang Chen, Erik Wallin, Erik Wallin, Guoxuan Xia, Miao Xiong, and Lars Hammarstrand for all the research discussions even though we don't have a publication yet. I truly appreciate your open-mindedness and willingness to exchange ideas. Thank you for fostering such a supportive and stimulating research environment. I also thank my master thesis supervisor, Alexandre Graell i Amat, for introducing me to the world of research, and to Jianan Liu and Lucas Rath for the research training I received at Huawei Research Center, Gothenburg.

I was fortunate to be part of Wallenberg Artificial Intelligence, Autonomous Systems and Software Program (WASP), funded by Knut and Alice Wallenberg Foundation. Thanks a lot for organizing all the amazing study trips, which have given me the opportunity to explore a variety of research topics beyond just reading papers.

Apart from research, I thank Jacob Klintberg, Yu Ge and Oguz for supporting my transition into a teaching assistant. A special thank to Anders Grauers for hosting the ISP meeting with patience and kindness. Certainly, I thank my past and present colleagues in the computer vision group including Torsten, Carl, Axel, James, Huu, José, Alex, Lucas, Ji, Rasmus, Ida, Georg, Kunal, David Nordström, Roman, David Nilsson, Christopher, Victor, Josef, Erik, Jennifer, Sofie, Yara, Richard, Bernardo, and Fredrik in the computer vision group for striving to create a better work environment. Wish you all

the best! I'd like to extend this gratitude to Fredrik Kahl for supporting the annual group retreat after Covid-19. It makes this Ph.D. journey more fun! I also want to thank our lovely administrators, Ann-Christine for ordering an Apple keyboard for me, and Natasha for ordering a white board although I almost forgot I made this request. I thank all the people who say hello to me every day at Chalmers, you make seventh floor more than a work place, and sixth floor more than a kitchen.

I would like to thank my friends, Guanqun Ni, Johanna Due, Jingjing Chen, Shuangshuang Chen, Hong Zhang, Yidong Chen, Yibo Wu, Ze Zhang, Yali Wang, Junjie Li, and Yu Ge for your company and support. Although we may not meet often, our discussions about life, career, research, and everything in between—often marked by differing opinions—have been incredibly enriching. Your sincerity and kindness have always pushed me to reflect more deeply and broaden my perspectives.

In the end, I would like to express my gratitude to my partner Huang. His belief in me has allowed me to stay true to myself, even in the most challenging moments. To my sister, Rui Liu, I always enjoy the discussions we have about science and engineering. To my father, thank you for pushing me to reflect on my decisions and encouraging me to strive for excellence. To my mom, thank you for always being there for me.

Acronyms

AD:	Anomaly Detection
AE:	Auto-Encoder
AUROC:	Area Under Receiver Operating Characteristic
BERT:	Bidirectional Encoder Representations from Transformer
BNNs:	Bayesian Neural Networks
BYOL:	Bootstrap Your Own Latent
CD:	Contrastive Decoding
CDF:	Cumulative Distribution Function
CE:	Calibration Error
CHAIR:	Caption Hallucination Assessment with Image Relevance
CLIP:	Contrastive Language-Image Pre-training
CP:	Conformal Prediction
DNNs:	Deep Neural Networks
EBMs:	Energy-based Models
ECE:	Expected Calibration Error
EMA:	Exponential Moving Average
FPR:	False Positive Rate
GAN:	Generative Adversarial Network
GAVIE:	GPT4-Assisted Visual Instruction Evaluation
HNNs:	Heteroscedastic Neural Networks
ID:	In-Distribution

JEM:	Joint Energy-based Modeling
KLD:	Kullback–Leibler Divergence
LLaVA:	Large Language and Vision Assistant
LLMs:	Large Language Models
LVLMs:	Large Vision-Language Models
MCE:	Maximum Calibration Error
MHSA:	Multi-Head Self Attention
MLP:	Multilayer Perceptron
MMCE:	Maximum Mean Calibration Error
MoCo:	Momentum Contrast
MSP:	Maximum Softmax Probability
NMT:	Neural Machine Translation
NTP:	Next Token Prediction
OE:	Outlier Exposure
OOD:	Out-of-Distribution
OSR:	Open Set Recognition
RAG:	Retrieval Augmentation Generation
SGLD:	Stochastic Gradient Langevin Dynamics
SigLIP:	Sigmoid Loss for Language-Image Pre-training
SM:	Score Matching
SSL:	Self-Supervised Learning
SSM:	Sliced Score Matching
SupCon:	Supervised Contrastive Learning
TPR:	True Positive Rate
TS:	Temperature Scaling

Contents

Part I

Overview

CHAPTER 1

Introduction

Over the past two decades, deep learning techniques have achieved tremendous success in a wide range of domains including computer vision, natural language processing, medical healthcare, autonomous systems, audio and speech processing, manufacturing industry, agriculture, and entertainment [1]. The emergence of ChatGPT [2] and GPT-4V(ision) [3] further demonstrates the great capability of these deep models. As of October 2024, over 180.5 million users have registered on ChatGPT, highlighting its widespread adoption and importance in the lives of people. Large vision-language models (LVLMs), such as GPT-4V(ision), further extend this capability to multi-modal understanding, enabling the model could process and interpret visual data along with text. For instance, GPT-4V(ision) can describe the iimages/scenes for the people who are visually impaired. Notably, Open-AI and Microsoft are developing applications such as "Be My Eye" [4] and "Seeing AI" [5] to assist the visually impaired. Moreover, LINGO-2 trained by WAYVE can be potentially integrated into autonomous driving to facilitate the human-vehicle interaction [6]. As importantly, vision-language models (VLMs) in medical domain [7] that accept both medical images and reports as inputs can be particularly beneficial for the patients with limited access to medical resources.

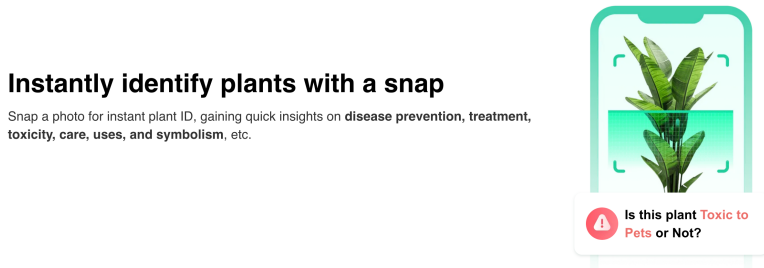

Figure 1.1: *One Use Case of OOD Detection.* The image is taken from the APP named PictureThis [11].

However, despite their success and potential, ensuring the reliability of such models when deploying in the real world is always a concern. The early concerns about artificial intelligence (AI) and machine learning (ML) risks are raised by Norbert Wiener back to 1960, who emphasizes the importance of designing systems that align with human intentions [8]. Dario Amodei and other researchers from OpenAI propose a seminal framework for ML safety in their paper "Concrete Problems in AI Safety" in 2016 [9], which outlines practical challenges such as robustness, out-of-distribution (OOD) detection, and error handling, establishing a foundation for targeted research in the domain of ML safety. As deep learning techniques have been widely utilized across various domains, ensuring the safety and reliability of deployed models has become more critical than ever.

In this thesis, the reliability of deep models is tackled through the lens of OOD detection, model calibration, and hallucination mitigation. To be specific, OOD detection reflects the ability that models know what they do not know [10]. For example, consider a deep model is trained to classify plant species (see Figure 1.1). If the model encounters a plant species it has not seen during training, it should not offer any treatment recommendations. Instead, the image should be referred to a botanist for examination and annotation. This approach not only enhances the accuracy of the application but also helps expand the database of plant categories. This ability is even more critical for high-stakes tasks such as medical image analysis and autonomous driving. In such scenarios, the diverse nature of the input data received after deployment can significantly impact the performance and behavior of deep models.

Moreover, it is unarguably that uncertainty of a prediction is also crucial for determining the reliability when deploying deep models in safety-critical real-world applications. For instance, a deep model predicts whether a patient has a particular disease based on medical imaging (*e.g.*, X-rays). If model A predicts "*Disease*" with a confidence score of 98% but the actual likelihood of the disease might only be 60%, this overconfidence could lead to unnecessary treatments or invasive procedures, causing harm or additional costs. If model B predicts "*Disease*" with a confidence score of 59%, which accurately reflects the uncertainty in the diagnosis, prompting doctors to order additional tests or seek second opinions before proceeding with treatment. In this case, model B is considered better calibrated, as a result, more preferable than model A.

Recently, large language models (LLMs) such as ChatGPT [2] and large vision-language models (LVLMs) such as GPT-4V(ision), also known as foundation models, suffer from the issue of hallucination, see OpenAI system card [3], [12] for more information. Hallucination in LLMs refers to the problem that either the output of LLMs is inconsistent with the source content in context, or LLMs generate responses that are not grounded by the pre-training dataset [13]. Not surprisingly, all VLMs are also affected by hallucinations because of the utilization of a language decoder. Here hallucination refers to the scenarios which VLMs occasionally generate responses that are not supported by the visual input, which can be catastrophic for the visually impaired and autonomous driving.

To this end, this thesis aims to enhance the reliability of deep models by focusing on three research problems that are briefly summarized as follows:

- OOD detection, *i.e.*, detecting semantically different inputs from the training in-distribution (ID) data.
- Model calibration, *i.e.*, improving the alignment between the accuracy and its associated confidence.
- Hallucination mitigation, *i.e.*, mitigating the object hallucination when the generated responses of VLMs are not grounded in the visual inputs.

1.1 Contributions

The thesis seeks to develop efficient and effective methods to enhance the reliability of deep learning systems. The key findings and main contributions

of this thesis are as follows:

- To enable and guarantee discriminative classifiers with enhanced OOD detection and better calibration, training within the framework of joint energy-based modeling (JEM), using sliced score matching, improves their ability to detect OOD data and results in better-calibrated classifiers (see Paper A).
- The concept of JEM devised for discriminative classifiers is transferred for regression tasks resulting in a better calibrated regressor (see Paper B).
- To design an effective OOD scoring method, an entropy-based OOD score is devised with access only to the information available in the probability space (see Paper C).
- To overcome the issue of data scarcity in medical domain, a lightweight feature-based framework is developed by leveraging transfer learning and self-supervised learning techniques such that medical anomalies can be detected without the need for reconstructing medical images or directly accessing annotated anomalies (see Paper D).
- To enable zero-shot OOD detection, knowledge gained from OOD detection within discriminative classifiers are successfully transferred to contrastive vision-language models (VLMs), extending the applicability of learned knowledge to a broader context and reducing the dependency on task-specific training ID data (see Paper E).
- To address object hallucination in generative VLMs, an energy-based decoding method is devised, drawing inspiration from insights gained from OOD detection (see Paper F).

1.2 Thesis outline

This thesis is divided into two parts. Part I consists of 5 chapters providing motivation, background and necessary methodologies regarding each application followed by the summary of papers and future work. Chapter 2 briefly outlines some fundamental knowledge regarding three learning paradigms of deep models and some methodologies on uncertainty estimation. Chapter 3

provides the preliminary knowledge regarding OOD detection, model calibration, and hallucination mitigation, respectively. Chapter 5 points out the direction of future work. Part II presents the detailed results of the included papers.

CHAPTER 2

Background

2.1 Deep models

Over the past two decades, deep learning-based models have progressed a lot, *i.e.*, starting from with only visual or textual input to several modalities (*e.g.*, images, language, video, and audio) as inputs. In this thesis, three representative models are considered and described below. A conceptual comparison is illustrated in Figure 2.1.

Discriminative vision models

Designing a better model configuration with generic and competitive visual capability has been extensively studied over years. The availability of ImageNet-1k [14] collected for classification tasks has greatly contributed to accelerating the development of deep models. There is a series of proposed deep neural networks (DNNs) consisting of convolutional neural networks [15], [16], residual neural networks [17]–[19], and transformer neural networks [20]–[22]. Meanwhile, devising effective loss functions is also critical for achieving superior visual recognition. Such losses include but not limited to cross-entropy

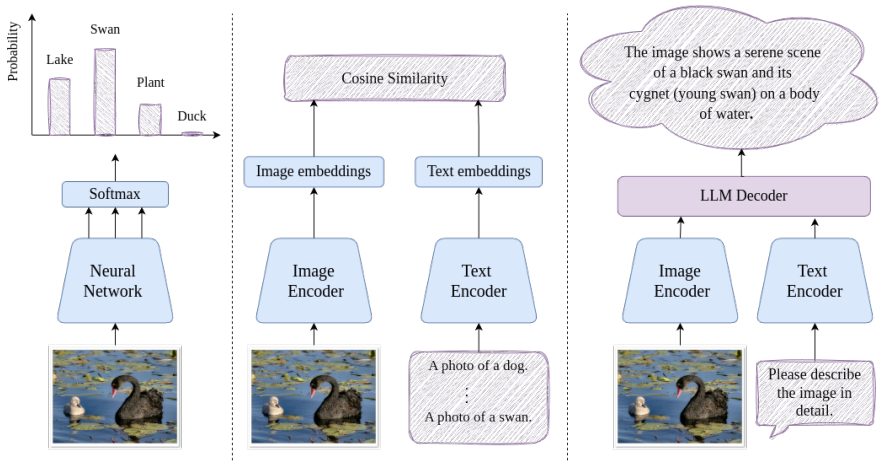

Figure 2.1: *A Conceptual Comparison of Three Representative Deep Models in the Inference Stage.* The three models include discriminative classifiers (left), contrastive vision-language models (center), and generative vision-language models (right).

loss for object classification, focal loss for imbalance datasets [23], and label smoothing [24]. In this thesis, models pretrained only with the cross-entropy loss (also known as discriminative classifiers) are mainly considered and are the main focus in Paper C. Mathematically, given a set of training data denoted by $D = \{\mathbf{x}_i, y_i\}_{i=1}^N$ with label space denoted by $\mathcal{Y} = \{1, 2, 3, \cdots, C\}$, a neural network parametrized by θ and denoted by $f(\mathbf{x}; \theta) : \mathcal{X} \to \mathbb{R}^C$ can be learned by minimizing the empirical risk, *i.e.*,

$$R_{\mathcal{L}}(f) = \mathbb{E}_D(\mathcal{L}_{\text{CE}}(f(\mathbf{x}; \theta), y)) \quad \text{and} \quad \mathcal{L}_{\text{CE}} = -\log \frac{\exp(f_y(\mathbf{x})/\tau)}{\sum_i^C \exp(f_i(\mathbf{x})/\tau)}, \quad (2.1)$$

where $f(\mathbf{x})$ is the output of the neural network, termed the logits, $f_y(\mathbf{x})$ denotes the logit corresponding to the ground-truth label y, and τ is the temperature.

Contrastive vision-language models

Contrastive Language-Image Pre-training (CLIP) [25] has recently received tremendous recognition because of its superior cross-modal alignment, *i.e.*, the alignment between text and image, which achieves competitive zero-shot classification accuracy compared to the supervised setting, more details can be found in [26]. A general CLIP-style architecture consists of a text encoder $\mathcal{T}$ and an image encoder $\mathcal{I}$. More than 400 million paired images and texts equipped with InfoNCE [27] enable its successful training. Specifically, given a mini-batch paired image-text data denoted by $\mathcal{B} = \{(\mathbf{x}_1, \mathbf{t}_1), \cdots, (\mathbf{x}_{|\mathcal{B}|}, \mathbf{t}_{|\mathcal{B}|})\}$, the training objective is to minimize

$$\mathcal{L}_{\text{CLIP}} = -\frac{1}{2|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \left(\log \frac{\exp(\tau \mathbf{f}_i \cdot \mathbf{g}_i)}{\sum_{j=1}^{|\mathcal{B}|} \exp(\tau \mathbf{f}_i \cdot \mathbf{g}_j)} + \log \frac{\exp(\tau \mathbf{f}_i \cdot \mathbf{g}_i)}{\sum_{j=1}^{|\mathcal{B}|} \exp(\tau \mathbf{f}_j \cdot \mathbf{g}_i)} \right), \tag{2.2}$$

where $\mathbf{f}_i = \frac{\mathcal{I}(\mathbf{x}_i)}{\|\mathcal{I}(\mathbf{x}_i)\|}$ and $\mathbf{g}_i = \frac{\mathcal{T}(\mathbf{t}_i)}{\|\mathcal{T}(\mathbf{t}_i)\|}$. The normalization in Eq. 2.2 has to be done for images and texts independently. Further, the scalar τ is parameterized as $\exp \tau'$ to ensure τ to be positive, where τ' is a global freely learnable parameter. SigLIP [28] is one of variations of CLIP trained with Sigmoid loss.

During inference, considering a dataset with label space denoted by $\mathcal{Y}_{\text{in}} = \{y_1, y_2, \cdots, y_C\}$, the default text prototype t_c for the class c can be constructed as `a photo of` $\langle y_c \rangle$, which is further processed by a text encoder. A test image $\mathbf{x}$ is firstly processed by an image encoder $\mathcal{I}$. The cosine similarity s_c between extracted feature $\mathcal{I}(\mathbf{x})$, and all text prototypes $\mathcal{T}(t_c)$ are taken as the logit, which is then normalized by Softmax. The probability of the image $\mathbf{x}$ belonging to class c can be calculated as

$$p_k(\mathbf{x}|\mathcal{Y}_{\text{in}}, \mathcal{I}, \mathcal{T}) = \frac{\exp(s_c/\tau)}{\sum_{j=1}^{C} \exp(s_j/\tau)}, \tag{2.3}$$

where $s_c = \frac{\mathcal{I}(\mathbf{x}) \cdot \mathcal{T}(t_c)}{\|\mathcal{I}(\mathbf{x})\| \cdot \|\mathcal{T}(t_c)\|}$, and τ is the temperature. This CLIP-style architecture is mainly considered in Paper E.

Generative vision-language models

Generative vision-language models (VLMs) such as GPT-4V(ision) attract much attention because it can generate responses for a given visual input.

For instance, they can be utilized to describe the real world to visually impaired people [4], [5]. Recent proposed VLMs have shown impressive performance on natural instruction-following and visual reasoning capabilities [29]–[32]. A general framework of VLMs consists of an image encoder, a text encoder, and a language decoder. LLaVA-1.5 [31] and InstructBLIP [32] are two representative VLMs and considered in Paper F. Specifically, LLaVA-1.5 [31] simply utilizes a Multilayer perceptron (MLP) layer to align the visual feature and text feature. InstructBLIP [32] employs the Q-former [33] to extract instruction-aware visual features from the output embeddings of the frozen image encoder. They both employ Vicuna-7B [34] as the language decoder. A comprehensive summary about recent VLMs can be found in [35]. The training of VLMs typically involves two-stage training including pre-training for feature alignment and fine-tuning with language-image instruction-following data. VLMs such as LLaVA [30], [31], are commonly trained in an autoregressive manner with a causal attention mask, meaning that the prediction of the current token x_t only depends on the previous tokens. Formally, given an image $\mathbf{X}_v$, an instruction $\mathbf{X}_{\text{instruct}}$, and target answers $\mathbf{X}_a$, the probability of target answers is defined as

$$p(\mathbf{X}_a|\mathbf{X}_v, \mathbf{X}_{\text{instruct}}) = \prod_{i=1}^{L} p_\theta(x_i|\mathbf{X}_v, \mathbf{X}_{\text{instruct},<i}, \mathbf{X}_{a,<i}), \tag{2.4}$$

where L is the length of target answer $\mathbf{X}_a$. During inference, the visual tokens denoted by $\mathbf{Z}_v \in \mathbb{R}^{N\times d}$ and textual tokens denoted by $\mathbf{Z}_{\text{instruct}} \in \mathbb{R}^{M\times d}$ are concatenated, and regarded as the final input tokens denoted by $\mathbf{Z} \in \mathbb{R}^{T\times d}$, to the language decoder. Further, the input tokens $\mathbf{Z}$ are processed by several transformer blocks, and each block consists of a multi-head self attention (MHSA) layer, layer normalization, and a feed forward layer as shown in Figure 2.2. The MHSA layer contains several parallel heads, and each head i is initialized with different keys $\mathbf{W}_i^K \in \mathbb{R}^{d\times d_k}$, values $\mathbf{W}_i^V \in \mathbb{R}^{d\times d_v}$, and queries $\mathbf{W}_i^Q \in \mathbb{R}^{d\times d_k}$. For each head i, the attention operation is defined as

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}\left(\frac{\mathbf{Q}\cdot\mathbf{K}^T}{\sqrt{d_k}}\right)\cdot\mathbf{V}, \quad \text{and} \tag{2.5}$$

$$\mathbf{Q} = \mathbf{Z}\cdot\mathbf{W}_i^Q, \quad \mathbf{K} = \mathbf{Z}\cdot\mathbf{W}_i^K, \quad \mathbf{V} = \mathbf{Z}\cdot\mathbf{W}_i^V. \tag{2.6}$$

Further, features obtained from each head are concatenated, formally,

$$\text{Multi-head}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \cdots \text{head}_s)\mathbf{W}^o \tag{2.7}$$

$$\text{where} \quad \text{head}_i = \text{Attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V), \tag{2.8}$$

where s is the number of heads, $d_k = d_v = d/s$, and $\mathbf{W}_i^o \in \mathbb{R}^{sd_v \times d}$. The obtained features are further processed by layer normalization and the feed forward layer. Finally, the output of the final block of language decoder is $\mathbf{h} = \{h_0, h_1, h_2, \cdots, h_{T-1}\}$. Implementation-wise, the last indexed hidden states denoted by h_t are commonly utilized for the subsequent next token prediction (NTP). To be specific, a learned vocabulary head $\mathcal{H}$ with the size of V_{size} is utilized to obtain the logits followed by Softmax to obtain a probability distribution of next token, formally,

$$p(x_t|x_{<t}) = \text{Softmax}[\mathcal{H}(h_t)], \tag{2.9}$$

where $x_{<t}$ denotes the sequence of tokens before t-th position $\{x_i\}_{i=0}^{t-1}$ and $\mathcal{H} \in \mathbb{R}^{d \times V_{\text{size}}}$.

Decoding mechanism After obtaining a probability distribution p of the next token over a fixed vocabulary $\mathcal{H}$, several decoding methods include but not limited to greedy decoding, top-p sampling (also known as nucleus sampling [36]), top-k sampling, and beam search decoding [37] can be utilized to generate token sequences. Each method is briefly reviewed below.

- Greedy decoding always selects the token (which can be a word, subword, or character) with the highest probability of all possible tokens in the model's dictionary at each step.
- Top-p decoding (also known as nucleus sampling [36]) draws samples from the minimal set of most probable tokens whose cumulative probability exceeds a threshold p.
- Top-k decoding restricts sampling to the top k tokens with the highest probabilities and re-normalizes their probabilities before drawing sampling.
- Direct sampling equals to nucleus sampling with $p = 1$.

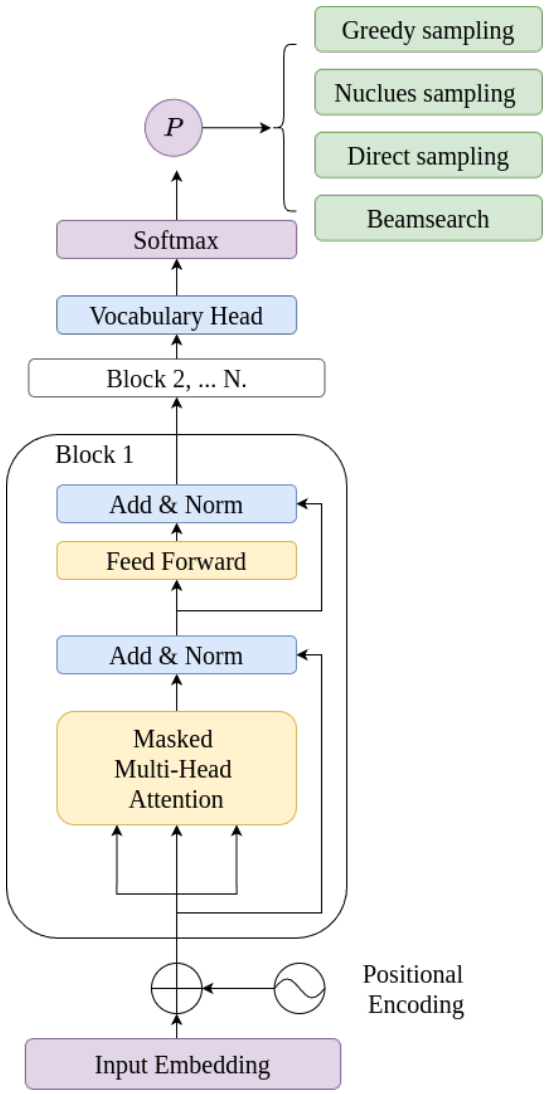

Figure 2.2: *Transformer-based Language Decoder Architecture.* The input embedding includes text embeddings and image embeddings.

- Beam search decoding explores multiple candidate sequences simultaneously and keep the top sequences with the hight probability (*i.e.*, the product of probabilities of all tokens in the sequences). The sequence with the highest probability is chosen as the final output sequence.

To summarize, greedy decoding and beam search are deterministic decoding methods, and beamsearch is more computational demanding compared to greedy decoding. Meanwhile, top-p, top-k and direct sampling belong to the category of stochastic methods due to the sampling operation.

2.2 Self-supervised learning

To make full use of unlabeled data, much effort has been put into devising self-supervised learning (SSL) loss functions (*e.g.*, contrastive loss [38], [39], reconstruction loss [40], clustering loss [41], similarity loss [42]–[44]) with different architectures (*e.g.*, Siamese network [38], [39], [45], student-teacher network [43], [46]) to facilitate the feature representation learning, which aims to obtain good transferable representation for various downstream tasks such as image classification, object detection, and semantic segmentation [38], [40]–[43], [45]–[47]. A conceptual comparison of three representative SSL methods including SimCLR [38], SimSiam [42], and DINO [43] is presented in Figure 2.3. Contrastive learning [48], one of the most popular learning paradigms, aims to construct such embedding space that features from similar samples stay close and features from dissimilar samples remain distant. By convention, similar (dissimilar) samples are referred to as positive (negative) pairs [1]. In practice, positive pairs can be samples that are two augmented versions of an input image. The rest of augmented samples at the same batch can be regarded as negative pairs. Generally, InfoNCE [27] is employed as the training loss. Such approaches commonly require negative pairs to avoid representation collapsing during the training stage such as MoCo [39], SimCLR [38], BarlowTwins [45]. That being said, such methods require a fairly large batch size. Instead BYOL [46], SimSiam [42], DINO [43], and DINOv2 [44] enable the successful training without utilizing the negative pairs while maintaining the competitive transferable representation for diverse downstream tasks. Nevertheless, [43], [44], [46] necessitate the use of the momentum encoder

[1] In the supervised setting, positive pairs involve samples that are different but share the same label information [49].

during training, which is computationally heavy and sensitive to hyperparameters. SimSiam [42] can be regarded as BYOL [46] without the momentum encoder. Therefore, self-training in a SimSiam manner [42] is considered in Paper D due to its simplicity and flexibility.

Training of SimSiam A brief description of its training process is provided below.

- An input image X is perturbed by sampling two different augmentations from the same augmentation distribution, denoted as $\mathcal{T}$, yielding X_1 and X_2;
- Further, two augmented samples are consecutively processed by a feature extractor f_θ, a projector h_ϕ and a predictor g_ψ, resulting in two features denoted by p_1 and Z_2;
- The goal is to minimize the negative cosine similarity between p_1 and Z_2, and the stop-gradient technique is employed to tackle the issue of feature collapsing. The final training loss is given as follows,

$$\mathcal{L}_{\text{SimSiam}} = \tfrac{1}{2}\mathcal{S}\big(p_1, \text{stopgrad}(z_2)\big) + \tfrac{1}{2}\mathcal{S}\big(p_2, \text{stopgrad}(z_1)\big), \tag{2.10}$$

  where $\mathcal{S}(p, z) = -\left\langle \frac{p}{\|p\|}, \frac{z}{\|z\|} \right\rangle$ is the negative cosine similarity.

2.3 Aleatoric uncertainty, epistemic uncertainty, and predictive uncertainties

Generally, the uncertainties existing in DNNs can be divided into data (aleatoric) uncertainty and model (epistemic) uncertainty [50]. Aleatoric uncertainty is caused by the noisy data such as the information loss about input samples due to the error and noise in the measurement systems. Epistemic uncertainty is caused by the pitfalls in the model such as training recipes, insufficient model structures, or lack of knowledge due to unknown samples, or a bad coverage of the training data [51]. Aleatoric uncertainty and epistemic uncertainty can be employed to induce predictive uncertainty, which is the confidence encapsulated in a prediction [52]. Within the framework of Bayesian neural networks

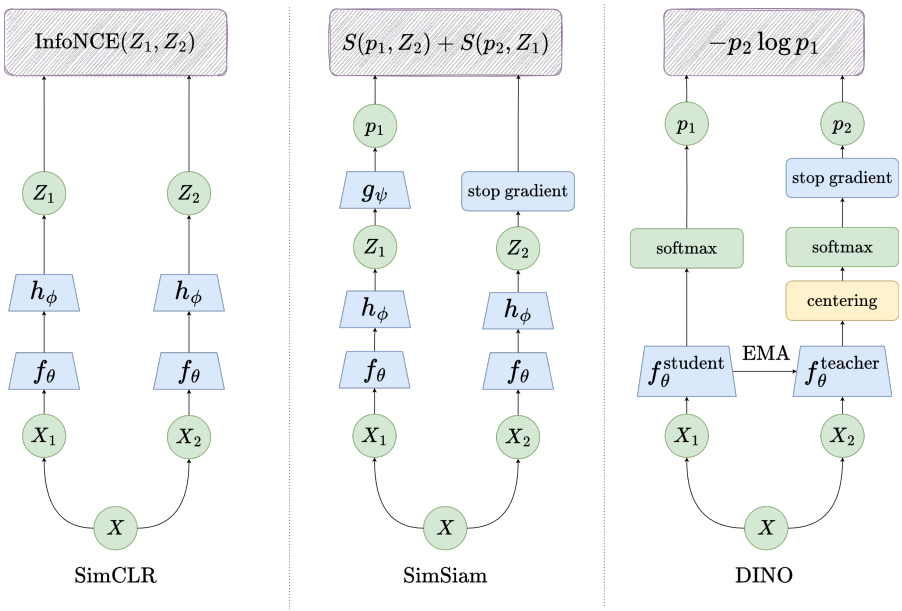

Figure 2.3: *A Conceptual Comparison of Three Representative SSL Methods.* From left to right, the three methods are SimCLR [38], Simsiam [42], and DINO [43]. A general framework of SSL consists of a image encoder f_θ, a projector h_ϕ, and a predictor g_ψ.

(BNNs), aleatoric uncertainty is estimated by placing a prior distribution over the output of the models, and epistemic uncertainty is captured by putting a prior distribution over the model parameters [50].

2.4 Proper scoring rule

We assume X is a random variable with realizations in $\mathcal{X}$, and $\mathcal{P}$ is a family of distributions over $\mathcal{X}$. A *scoring rule* is a penalty function denoted by $S(\mathbf{x}Q)$ and it measures the quality of a reference distribution Q for $\mathbf{x}$.

> **Definition 1** The scoring rule S is proper with respect to $\mathcal{P}$ if, for $P, Q \in \mathcal{P}$, the expected score $S(P, Q)$ is minimized in Q at $Q = P$. Further S is strictly proper if this is the unique minium: $S(P, Q) > S(P, P)$ for $Q \neq P$ [53].

Equivalently, the associated divergence or discrepancy between two distributions P and Q is defined as $D(P, Q) := S(P, Q) - S(P, P)$ and it is always non-negative for a proper score rule $S(\cdot, \cdot)$. Denoting the density of the distribution of Q as $\mathbf{q}(\cdot)$ and considering the proper scoring rule to be the *log score*, *i.e.*, $S(\mathbf{x}, Q) = -\log \mathbf{q}(\mathbf{x})$, the associated divergence between P and Q is the Kullback–Leibler divergence (KLD). Similarly, constructing the proper scoring rule as

$$S(\mathbf{x}, Q) = G(\mathbf{q}) + \langle \nabla G(\mathbf{q}), \mathbf{x} - \mathbf{q} \rangle, \tag{2.11}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product, G is a differentiable and concave function, and the associated divergence is a Bregman divergence.

Bregman divergence

Bregman divergence $D_G(\mathbf{p} \| \mathbf{q})$ between 2 elements $\mathbf{p}, \mathbf{q} \in \Delta^C$ can also be interpreted as the *linearization error* resulting from the linear approximation of G at $\mathbf{p}$, evaluated at $\mathbf{q}$. Mathematically, it is defined as

$$D_G(\mathbf{p} \| \mathbf{q}) := G(\mathbf{q}) - G(\mathbf{p}) + (\mathbf{p} - \mathbf{q})^\top \nabla G(\mathbf{q}), \tag{2.12}$$

which is non-negative for concave G. Here G is a differentiable and concave function on the space of categorical distributions Δ^C. C denotes the number of

classes, $\mathbf{p}$ is a one-hot vector, and the function G can be either the Shannon entropy or "gamma entropy", both of which are employed in Paper C for detecting OOD samples.

Unnormalized density estimation

Accurately estimating data density could directly contribute to detecting OOD samples based on their estimated density. Intuitively, samples with higher density are considered ID samples, while those with lower density are considered OOD samples. Energy-based models (EBMs) are one of the approaches utilized to estimate data density.

Energy-based models For a data point $\mathbf{x}$, its associated probability distribution can be represented as

$$p_\theta(\mathbf{x}) = \frac{\exp\left(-E_\theta(\mathbf{x})\right)}{Z_\theta}, \quad Z_\theta = \int \exp\left(-E_\theta(\mathbf{x})\right) d\mathbf{x}, \tag{2.13}$$

where θ is model parameters, $E_\theta(\mathbf{x})$ refers to energy function, and Z_θ is the partition function. Score matching [54] and noise contrastive estimation [55] are two fundamental methods suitable for learning unnormalised probabilistic models. In this thesis, score matching [54], particularly, sliced score matching [56] is mainly considered in Paper A and Paper B.

Score matching Intuitively, score matching learns the unnormalised models by minimizing the squared distance between the *score functions* (*i.e.*, the gradients of the log-density, $\nabla_{\mathbf{x}} \log p(\mathbf{x})$) of the data and the model distribution, p_d and p_θ, respectively. The partition function of the model distribution does not appear in the objective (due to the derivative). Mathematically,

$$J_{\mathrm{SM}}(\theta) = \mathbb{E}_{\mathbf{x}\sim p_d} \tfrac{1}{2} \left[\|\nabla_{\mathbf{x}} \log p_d(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x})\|_2^2 \right]. \tag{2.14}$$

Sliced score matching A recent extension of score matching is *sliced score matching* [56], which significantly improves the computational costs of score matching for high-dimensional input spaces and is therefore considered in Paper A and Paper B. To be specific, sliced score matching [56] replaces the (vectorial) score function by respective projections onto random directions. In

particular, the variance-reduced version of sliced score matching is utilized in Paper A and Paper B. Formally,

$$J_{\mathrm{SSM}}(\theta) = \mathbb{E}_{\mathbf{x}\sim p_d, \mathbf{v}\sim p_v} \tfrac{1}{2} \|\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x})\|^2 + \mathbf{v}^\top (\Delta \log p_\theta(\mathbf{x}))\mathbf{v}, \tag{2.15}$$

where Δ is the Laplace operator ($\Delta \coloneqq \sum_i \frac{\partial^2}{\partial x_i^2}$) and p_v is a radially symmetric distribution (*e.g.*, a multivariate standard normal distribution and a multivariate Rademacher distribution) to generate random directions. The finite sample version $\tilde{J}_{SSM}$ of J_{SSM} is obtained by averaging over the training set and by continuously sampling directions from p_v.

CHAPTER 3

Model Reliability

The reliability of deep models is a central topic of this thesis. This chapter examines their reliability through applications in OOD detection, model calibration, and hallucination mitigation. It begins with formal definitions and evaluation metrics, followed by a detailed review of recently proposed methods for each application. Related research problems for each application are also briefly mentioned to provide a more comprehensive perspective.

3.1 OOD detection

The terminology of OOD detection can be interpreted differently depending on different tasks and domains. Thanks to a recent survey paper [61], it clarifies and unifies the usage of various terminologies such as anomaly detection (AD), novelty detection, OOD detection, open-set recognition (OSR), and outlier detection. In this thesis, I mainly focus on the tasks of AD and OOD detection. Specifically, AD refers to detecting the samples that are different from the ID data, without considering the specific class labels within the ID

[1]Image source: ViM [60].

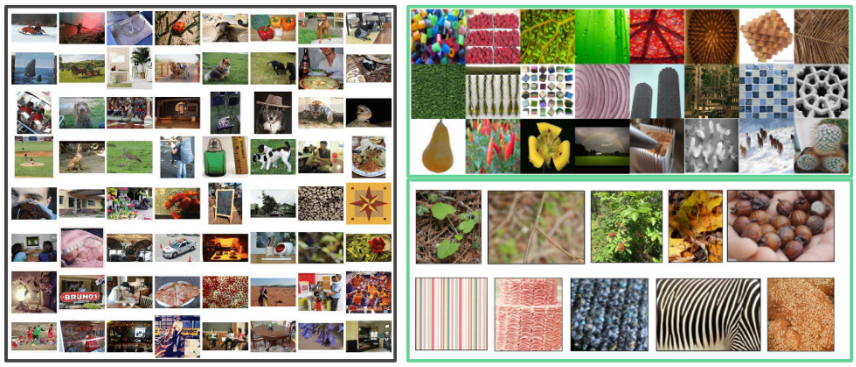

Figure 3.1: *Representative Samples from Large-Scale ImageNet-1k OOD Detection Benchmark*[1]. The left plot highlighted in black is in-distribution (ID) dataset, *i.e.*, ImageNet-1k [14]. The plots on the right, highlighted in green, represent out-of-distribution (OOD) datasets. From top to bottom, these include OpenImage-O [57], iNaturalist [58], and Textures [59].

data. However, OOD detection, considering a multi-class classification problem, refers to detecting any samples that are semantically different from the ID data and classifying the ID data correctly. For instance, if the ID dataset is ImageNet-1k [14], the corresponding OOD datasets could be OpenImage-O [57], Textures [59], and iNaturalist [58]. Representative samples from these datasets are shown in Figure 3.1. In this thesis, AD is addressed in Paper D, and OOD detection is tackled in papers A, C, and E.

Generally, OOD detection can be formulated as a binary classification problem, and the goal is to learn a score $s_\theta(\mathbf{x}) \in \mathbb{R}$. Mathematically, a binary classifier $g_\tau(\mathbf{x})$ is defined as

$$g_\tau(\mathbf{x}) = \begin{cases} \text{in}, & s_\theta(\mathbf{x}) > \tau, \\ \text{out}, & s_\theta(\mathbf{x}) \leq \tau, \end{cases} \tag{3.1}$$

where θ and τ are the model weights and the threshold, respectively. Two commonly utilized evaluation metrics include area under the receiver operating characteristic curve (AUROC), which evaluates the *average* performance of designed score function $s_\theta(\mathbf{x})$ by choosing different threshold τ, and FPR95—

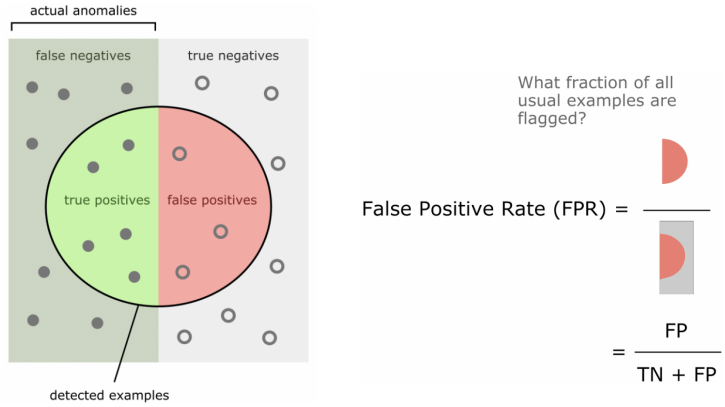

Figure 3.2: *Conceptual Visualization of the False Positive Rate (FPR)*[2]. FP denotes the number of false positives, and TN denotes the number of true negatives.

the false positive rate when the true positive rate is 95%. The conceptual visualization of FPR and AUROC can be found in Figure 3.2 and 3.3, respectively.

In this section, I first introduce the methods developed for OOD detection using visual-only backbones. Subsequently, the techniques proposed for visual-language models are presented.

Vision-based OOD detection

At the early stage of deep learning, the utilized backbones are single-modality, *i.e.*, with only visual input. Considering the utilized datasets, architecture configurations along with the training losses, the methods proposed for OOD detection can be roughly categorized into four groups:

- classification-based methods, utilizing cross-entropy loss as the primary loss function.

[2]Image source: https://hendrycks.github.io
[3]Image source: Drawn by CMG Lee based on http://commons.wikimedia.org/wiki/File:roc-draft-xkcd-style.svg

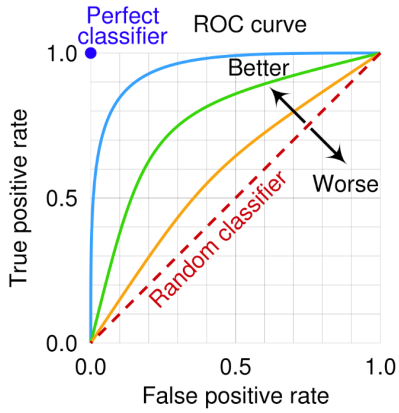

Figure 3.3: *Conceptual Visualization of the Area Under the Receiver Operating Characteristic Curve (AUROC)*[3]. The ROC curve illustrates the relation between the true positive rate (TPR) and the false positive rate (FPR) values at different thresholds. TPR $= \frac{\text{TP}}{\text{FN}+\text{TP}}$, where TP denotes the number of true positives, and FN denotes the number of false negatives.

- distance-based methods that learn compact ID feature representation utilizing self-supervised or deep metric learning;
- reconstruction-based methods that can easily be distinguished based on the name, *e.g.*, the reconstruction loss utilized in auto-encoder (AE) or generative adversarial network (GAN);
- density-based methods that learn the ID data distribution.

Classification-based methods

A good classifier is commonly trained with cross-entropy loss along with regularization losses such as label smoothing [24] or Mixup [62] to obtain better classification accuracy. Nevertheless, particularly, in Paper C, the classifier trained with only cross-entropy loss is considered. Methods relying on a discriminative classifier further can be divided into three categories including score design, enhancing methods, and training loss modification.

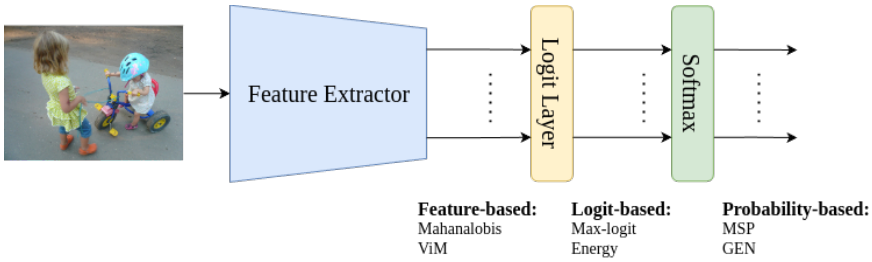

Figure 3.4: *Illustration of Classification-based OOD Scoring Methods.* OOD scoring methods are categorized based on the information they utilize at different stages: feature, logit, and probability.

Score design Given a classifier trained with cross-entropy loss and an input image $\mathbf{x}$, such approaches [60], [63]–[73] aim to design a suitable score function $s_\theta(\mathbf{x})$ for distinguishing between ID and OOD data accurately. They either rely on the information from the feature space [60], [69], [71]–[73], or from the logit space [65], [66], or from the probability space [63], [66] depending on the information utilized at which stage. An illustrative example showing the OOD scoring methods is shown in Figure 3.4, and a comprehensive technical comparisons is summarized in Table 3.1. More detailed discussions can be founded in [61].

Enhancing methods A plethora of research focuses on enhancing the OOD detection performance for given score functions [74]–[77]. Intuitively, they reshape the feature representation by either modifying the input samples [74], [75], or clipping the features at the penultimate layer [76], or removing specific feature information [77], or rescaling the feature from the penultimate layer [78], [79] to further separate the features extracted from ID and from OOD data. A detailed comparison among different enhancing methods is summarized in Table 3.2. ReAct [76] is commonly utilized in practice because of its simplicity and effectiveness.

Training loss modification Such methods typically concern with additional loss terms [81]–[88], or accessing outliers[89], [90], or synthesizing outliers using ID embeddings [91]–[93]. [81] adds a separate head after the penultimate

Methods	Equation	Free of			Space	
		ID train data	ID label	features	logit	probability
MSP [63]	$\max_c p_c$	✓	✓			✓
MaxLogit [66]	$\max_c f_c(\mathbf{z})$	✓	✓		✓	
KL-Matching [66]	$-\min_c D_{\mathrm{KL}}(\mathbf{p}\|\|\bar{\mathbf{p}}_c)$	✗, $\bar{\mathbf{p}}_c$	✗			✓
Energy [65]	$\mathrm{LogSumExp}\, f(\mathbf{z})$	✓	✓		✓	
NuSA [69]	$\frac{\sqrt{\|\mathbf{z}\|^2-\|\mathbf{z}^{w^\perp}\|^2}}{\|\mathbf{x}\|}$	✓	✓	✓		
Mahalanobis [64]	$\max_c -(f(\mathbf{z})-\hat{\mu}_c)\hat{\Sigma}^{-1}(f(\mathbf{z})-\hat{\mu}_c)$	✗, $\hat{\Sigma}_c, \hat{\mu}_c$	✗	✓		
GradNorm [67]	$\|\mathbf{p}-\mathbf{1}/C\|_1 \cdot \|\mathbf{z}\|_1$	✓	✓	✓		✓
pNML [68]	$\log\sum_{c=1}^{C} \frac{p_c}{p_c+p_c^\kappa(1-p_c)}, \quad \kappa=\frac{\mathbf{z}^\top\Sigma_{\mathrm{corr}}\mathbf{z}}{1+\mathbf{z}^\top\Sigma_{\mathrm{corr}}\mathbf{z}}$	✗, Σ_{corr}	✓	✓		
Residual [60]	$-\|\mathbf{z}^{P^\perp}\|_2$	✗, P	✓	✓		
ViM [60]	$-\alpha\|\mathbf{z}^{P^\perp}\|_2+\mathrm{LogSumExp}\, f(\mathbf{z})$	✗, α, P	✓	✓	✓	
NN-guide [71]	$\mathrm{Energy}(\mathbf{z})\cdot G(\mathbf{z}), \quad G(\mathbf{z})=\frac{1}{k}\sum_1^k s_{(i)}\mathrm{sim}(\mathbf{z}_{(i)},\mathbf{z})$	✗	✓	✓	✓	
NECO [72]	$\frac{\|P\mathbf{z}\|}{\|\mathbf{z}\|}\cdot\max_c f_c(\mathbf{z})$	✗, P	✓	✓	✓	
FDBD [73]	$\frac{1}{\|\mathcal{C}\|-1}\sum_{c\in\mathcal{C}\setminus\hat{c}}\frac{\tilde{D}_f(\hat{c},c)}{\|\mathbf{z}-\mu_{\mathrm{train}}\|}, \quad \tilde{D}_f(\hat{c},c):=\frac{\|f_{\hat{c}}(\mathbf{z})-f_c(\mathbf{z})\|}{\|\mathbf{w}_{\hat{c}}-\mathbf{w}_c\|}$	✗, μ_{train}	✗	✓	✓	
Shannon [70]	$-\sum_{m=1}^{M} p_{i_m}\log p_{i_m}, \quad p_{i_1}\geq\cdots\geq p_{i_C}, \ \gamma\in(0,1)$	✓	✓			✓
GEN [70]	$G_\gamma(\mathbf{p})=-\sum_{m=1}^{M} p_{i_m}^\gamma(1-p_{i_m})^\gamma, \quad p_{i_1}\geq\cdots\geq p_{i_C}, \ \gamma\in(0,1)$	✓	✓			✓

Table 3.1: *Taxonomy Comparison of OOD Scoring Methods.* $\mathbf{x}$ is an input, $\mathbf{z}$ is the feature extracted from the penultimate layer, $f(\mathbf{z})$ denotes logits, $\mathbf{p}=\mathrm{Softmax}(f(\mathbf{z}))$ is predictive distribution, $\bar{\mathbf{p}}_c$ is the per-class predictive distribution, P is the principle space spanned by eigenvectors of the largest D eigenvalues of the covariance of training data, $\hat{c}$ is the predicted class, and C is the number of classes. $\hat{\mu}_c$ and $\hat{\Sigma}_c$ are the empirical mean and covariance matrix of feature obtained training data belonging to class c, respectively. Σ_{corr} is the correlation matrix of features extracted from training data. μ_{train} is the empirical mean features across all training data.

Methods	Equation	Free of		Compatible with	
		ID train data	OOD data	MSP	Energy
ODIN [74]	$\tilde{\mathbf{x}} = \mathbf{x} + \varepsilon \text{sign}\left(\nabla_{\mathbf{x}} \log \max_c p_c(\mathbf{x})\right)$	✓	✗	✓	
ReAct [76]	$\tilde{\mathbf{z}} = \min(\mathbf{z}, b)$	✗, b	✓	✓	✓
RankFeat [77]	$\tilde{\mathbf{o}} = \mathbf{o} - s_1 \mathbf{u}_1 \mathbf{v}_1^\top$	✓	✓		✓
ASH [80]	$\tilde{\mathbf{z}} = \mathbf{W} \cdot (\mathbf{z} \circ s_f(\mathbf{z})) + \mathbf{b}$, where $s_f(\mathbf{z})_j = \begin{cases} 0 & \text{if } \mathbf{z}_j \leq P_p(\mathbf{z}), \\ \exp(r(\mathbf{z})) & \text{if } \mathbf{z}_j > P_p(\mathbf{z}). \end{cases}$	✓	✓	✓	✓
SCALE [79]	$\tilde{\mathbf{z}} = \mathbf{W} \cdot (\mathbf{z} \circ s_f(\mathbf{z})) + \mathbf{b}$, where $s_f(\mathbf{z})_j = \exp(r(\mathbf{z}))$	✓	✓	✓	✓

Table 3.2: *Taxonomy Comparison of Enhancing Methods.* $\mathbf{x}$ is an input, $\mathbf{z}$ is the feature extracted from the penultimate layer, $\mathbf{o}$ is the feature extracted from the intermediate layer, $f(\mathbf{z})$ denotes logits, $r(\mathbf{z})$ is a scaling factor defined as the ratio of sum of all activations versus sum of unpruned activation in $\mathbf{z}$, *i.e.*, $r(\mathbf{z}) = \frac{Q(\mathbf{z})}{Q_p(\mathbf{z})}$, where $Q(\mathbf{z}) = \sum_j \mathbf{z}_j$, and $Q_p(\mathbf{z}) = \sum_{\mathbf{z}_j > P_p(\mathbf{z})} \mathbf{z}_j$, $P_p(\mathbf{z})$ is defined as the value of p^{th} percentile of the elements in feature $\mathbf{z}$.

layer of the original network. This confidence branch can be one or more fully-connected layers followed by a sigmoid normalization. To be specific, if the prediction is confident, the output of confidence should be closer to 1 otherwise 0. Moreover, the final probability distribution is obtained by interpolating between the original predictions and the targeted probability distribution. The degree of interpolation is indicated by the predicted confidence. Additionally, to prevent the network from minimizing the loss by always choosing the ground truth labels, a log penalty term is added to encourage the network to always be confident. Unlike [81] that modifies the network architecture, [82], [83] simply reinterpret logits as joint log-probabilities (over inputs and labels). Further, the model is trained using a log-evidence term and the standard cross-entropy loss, leading to improved OOD detection performance. [85] operates with the assumption that the first singular vector of the autocorrelation matrix is a robust mean estimator. Specifically, its training includes 1) initializing the weight matrix $\mathbf{w}$ with orthonormal vectors and freezing them during the training; 2) the logit of each class is calculated as the absolute value of the cosine similarity between the feature of a test sample and the weight matrix corresponding to the class c, *i.e.*, $\mathbf{w}_c$. During inference, the utilized OOD score is defined as the minimum angular distance between the test feature and the first singular vector of each class. [84] decomposes the semantic labels into eight different groups based on the WordNet [94] and introduces one category "others" to each group. During training, the category of "others" is taken as

the ground truth class within the groups that the ground-truth class c is not included. The training loss is the sum of cross-entropy loss across each group. The OOD score is defined as the lowest probability assigned to the "Other" class across all groups. [87] and LogitNorm [88] share a similar idea, *i.e.*, normalizing the logit vector to a unit vector before applying Softmax. However, the difference is [87] uses the maximum cosine similarity between the feature and $\mathbf{w}_c$ as the OOD score, and the value of temperature is inferred by batch normalization. LogitNorm [88] employs the maximum softmax probability as the OOD score.

To better learn the decision boundary between ID and OOD data, [89], [90] incorporate OOD samples into the training procedure. [89] aims to minimize the cross-entropy loss for ID samples and the KL divergence between the uniform distribution and the predictive distribution for OOD samples. OECC [90] proposes to minimize the squared distance between the training accuracy and the average confidence in its predictions for the ID samples, as well as the total variation distance between the uniform distribution and the predictive distribution produced by OOD samples, while also minimizing the cross-entropy loss. Although outlier exposure (OE) could improve the performance of OOD detection by a notable margin compared to the training with only ID data, the dependence on outlier data makes it less favorable in practice because of the intensive effort required to collect curated OOD data. To resolve this issue, [86] employs GAN to generate OOD samples. To be specific, the generated samples are forced to produce a uniform distribution given a classifier. Meanwhile, the discriminator is trained considering the generated samples are fake and ID samples are real. VOS [92], NPOS [91], and DreamOOD [93] instead turn to synthesize virtual outliers from the ID feature representation. Specifically, [92] regards the class-wise ID features as the multivariate distribution, where the mean and covariance can be calculated from the training data sharing the same class. Subsequently, the virtual outliers are generated by sampling the low-density region of the estimated class-conditional distribution. Finally, a non-linear MLP layer is added to reshape the energy landscape by minimizing the energy for ID data and maximizing the energy for the virtual OOD data. NPOS [91] adopts a similar principle but in a non-parametric manner. It utilizes k-nearest neighbor (k-NN) to filter the boundary samples. Further, only the highest portion of boundary samples is utilized to synthesize OOD samples. The training loss is

the same as VOS [92]. DreamOOD [93] employs a similar learning principle as VOS [92]. The difference is that DreamOOD [93] harnesses the power of diffusion models. First, the class-wise text embeddings extracted from the CLIP encoder are utilized as the class prototypes during training. Afterwards, virtual outliers are synthesized based on the learned, text-conditioned training feature and further processed by Stable Diffusion [95]. Compared to VOS [92] and NPOS [91], the OOD samples are generated in pixel space. Finally, the same energy regularization loss as VOS [92] is applied. Although diffusion models are powerful enough to generate realistic OOD samples, generating outliers in pixel-space is fairly expensive.

Reconstruction-based methods

The assumption made for reconstruction-based method is that ID samples tend to produce lower reconstruction errors compared to OOD samples. Such approaches are preferred in medical anomaly detection [96]–[100]. The reasons are two-fold. First, the anomalies in medical imaging are commonly rare while the training of reconstruction methods only require to access the normal samples. Second, the abnormal regions are likely to produce higher reconstruction error, which is beneficial to localize the abnormal regions. Several reconstruction-based methods are devised for anomaly detection, *e.g.*, auto-encoders and their variants [96], [97], [101]–[104], and generative adversarial networks (GANs) such as f-AnoGAN [98]. Recently, diffusion models and their variants have received significant attention because of their powerful mode coverage over GANs [99], [100] and their ability to generate more realistic sample quality compared to variational autoencoders (VAEs). However, such approaches heavily rely on massive amounts of training data resulting in a high computational load, and are therefore less favorable in practice.

Distance-based methods

Distance-based methods rely on the assumption that the feature extracted from OOD samples is far away from ID samples [64], [105]–[107]. Such methods are quite flexible to be utilized regardless of the type of loss functions [64], [107]. Mahalanobis distance and k-nearest neighbor (k-NN) are two standard methods in practice [64], [105], [106]. Specifically, Mahalanobis distance [63] assumes a Gaussian distribution for the class-wise feature embedding, im-

plying that the label information of the training samples is accessible. Such assumption is not always correct, and the label information of the training data might not be accessible. Unlike Mahalanobis distance, k-NN is quite appealing when the label information is not accessible [105], [106]. For instance, [105] combines the off-the-shelf generic feature extractor (*e.g.*, ResNet trained on ImageNet-1k) with k-NN for anomaly detection. [106] utilizes k-NN for OOD detection given a pre-trained classifier, replacing cross-entropy loss with supervised contrastive learning (SupCon) loss, which results in better OOD performance. Overall, k-NN works well for the case when OOD samples lie on the manifold far from the ID samples. However, due to the symmetries of k-NN, it might not be able to detect OOD samples that are semantically close to the ID samples as pointed out in VGLR [108]. Therefore, VGLR proposes a likelihood ratio-based [109] OOD score utilizing k-NN, considering the geometry of data around the nearest neighbor and irrelevant "background features". Under the constraint of limited annotated training data (*i.e.*, Chest X-rays images in our case), in Paper D, we propose a light weight training with accessing limited ID data trained with SimSiam [42] for anomaly detection [110]. The resulting framework, employing k-NN algorithm as the OOD score, is data efficient and robust to outliers in training data.

Density-based methods

Such methods [83], [104], [109] rely on the assumption that ID samples lie on the region with high density. Therefore, they focus on learning a good ID data density estimator. Pioneering works include GMM [104], which models ID data using a Gaussian mixture model, and JEM [82], [83], which learns the data distribution through energy-based modeling. Likelihood ratio [109] suggests training an auto-regressive *semantic* model at the pixel level in image space to estimate the ID data distribution. To decouple the irrelevant background information, a *background* model is also trained in a auto-regressive manner by adding perturbations to the input data and randomly selecting pixels following an independent and identical Bernoulli distribution. The final OOD score is devised as the likelihood ratio between the image model and the background model. Normalizing flows (NFs) are a promising approach for modeling data distributions, as they provide exact likelihood estimation. However, empirical studies have shown that NFs tend to be overconfident in detecting OOD samples [111], [112]. [111] proposes to analyze

this phenomenon by linearizing the difference in expected log-likelihoods (*i.e.*, $\mathbb{E}_q(\log p(\mathbf{x},\theta)) - \mathbb{E}_{p^*}(\log p(\mathbf{x},\theta))$[4]). It suggests against using density estimates from NFs for OOD detection until their estimates for OOD samples are better understood. [112] reveals that NFs tend to learn the local pixel correlations and generic transformation from image to latent space, rather than learning the semantic structure of the ID data. While density-based methods are theoretically appealing, they generally perform less competitively than classification-based approaches.

Vision-language based OOD detection

CLIP [25] is getting recognition for the task of OOD detection because of its superior alignment between image and text [93], [113]–[118]. [113] is the first work to explore the capability of CLIP for zero-shot OOD detection via manually constructing an OOD label set denoted by $\mathcal{Y}_{\text{OOD}}$. Clearly, this restricts its deployed scenarios because it requires to construct the OOD label set for different ID datasets. To resolve the inconvenience of manually designing OOD labels, as discussed in [113], ZOC [114] instead trains a text description generator to obtain $\mathcal{Y}_{\text{OOD}}$ automatically. NegLabel [118] devises an effective algorithm to select a set of OOD labels via exploiting the lexical database such as WordNet [94] given the ID labels. The resulting score function is defined as the ratio of the exponential sum of all ID logits to the combined exponential sum of both ID logits and OOD logits. Although [113], [114] demonstrate superior performance on OOD detection, they both rely on pre-defined OOD label sets, which unavoidably impedes their performance as the defined OOD labels might deviate from the real OOD label. Furthermore, the OOD label set potentially has to be collected for every ID dataset. Instead, CLIPN [116] fine-tunes the CLIP [25] by introducing an additional text encoder on par with negative (learnable) prompts. Similar principles have also been explored in [117]. However, the fine-tuning of CLIP [25] inevitably has to be done for each ID dataset. MCM [115] instead neither depends on the design of the OOD label nor requires additional fine-tuning. It directly uses the text embeddings processed from the prompts `this is a photo of a` $\langle y_c \rangle$ as the concept prototypes to perform OOD detection. Our method TAG proposed in

[4] $p(\mathbf{x},\theta)$ is a generative model, p^* is the training data distribution, q is some dissimilar distribution with support on $\mathcal{X}$, and has a higher likelihood compared to the distribution of training data.

Paper E requires neither pre-defined OOD labels nor pre-training. Moreover, it can be applied to MCM [115], potentially enhancing the performance of OOD detection.

Related research problems

There are some research problems that are closely related to the OOD detection including but not limited to semantically coherent OOD detection [119]–[121], open-set semi supervised learning [122], [123], and selective classification with OOD detection [124], [125].

Semantically coherent OOD detection (SC-OOD) Most existing benchmarks developed for OOD detection simply consider one dataset (*e.g.*, ImageNet-1k) as the ID dataset and other datasets (*e.g.*, Texture and ImageNet-O) as the OOD datasets. It is true that such benchmarks could fairly accurately reflect the reliability of the models detecting samples with semantic shift. However, it ignores the case where samples share the same semantic class but from another datasets. For instance, a model trained on ImageNet-1k is expected to correctly classify images of a cat from both ImageNet-1k and CIFAR-100. That is to say, a good classifier is expected to detect samples with semantic shift as well as robust to samples with covariate shift. SC-OOD [119] empirically shows that post-hoc scores such as Energy [65] encounter performance degradation in such scenarios, particularly, in terms of FPR95. A similar work [121] aims to resolve this issue by decomposing the overall feature of an input into invariant (*i.e.*, the essential feature to decide semantic labels) and environmental feature (*i.e.*, non-invariant features) components. It empirically reveals a similar phenomenon; existing OOD scores are sensitive to the environmental features (*e.g.*, background/style), meaning that they might struggle to detect samples with the same semantic class but different environmental conditions. Therefore, SEM [126] constructs three benchmarks that consider both the detection of semantic shift samples and robustness to covariate shift samples. Meanwhile, it also proposes an OOD score based on both low-level and high-level features.

Open-set semi-supervised learning The devised OOD scores such as MSP [63] and Energy [65] score are beneficial in the framework of open-set semi-supervised

learning [122], [123]. To be specific, the score function can be utilized to identify unlabeled samples likely belonging to ID data, which are then *pseudo-labeled* and treated as labeled training data.

Selective classification with OOD detection (SCOD) [125] firstly proposes to unify the tasks of OOD detection and misclassification, *i.e.*, a reliably deployed model is expected to detect or reject samples that are either OOD samples or misclassified ID samples. [124] further reveals that existing OOD scores are not as effective as detecting misclassified ID samples compared to the tasks of detecting OOD samples.

3.2 Model calibration

When deploying deep models, one might care not only about the correctness of the prediction but also the corresponding confidence, particularly, for safety-critical applications such as autonomous driving and medical imaging diagnosis. For example, a self-driving car using a classifier to detect objects (*e.g.*, pedestrians, traffic signs, and lanes) should rely on other sensors when the camera-based prediction confidence is low. Similarly, medical doctors should manually verify diagnoses with low confidence. Therefore, one would expect that the confidence in a prediction should reflect the probability that it is correct.

From the Bayesian perspective, accurately estimating model uncertainty (see Section 1) is essential to obtain a well-calibrated model. Bayesian neural networks (BNNs) are a classic way to capture the model uncertainty by putting a prior distribution over the model parameters [127], [128]. However, such approaches require Bayesian inference, which is fairly challenging because of the average over all possible weights (referred to as *marginalization*). In practice, MC-dropout [129] is a commonly employed method to estimate model uncertainty because of its simplicity, *i.e.*, training a model with dropout applied before every weight layer and performing dropout during inference to sample from the approximate posterior [50]. Apart from that, heteroscedastic neural networks (HNNs) [50] are also considered because they can model the aleatoric uncertainty and epistemic uncertainty jointly. Deep ensembles [130] are preferable in real applications because of implementation simplicity. Bayesian approaches generally work well for both classification and

regression tasks. In the following section, the calibration methods designed for classification and regression tasks are briefly summarized.

Calibration for classification

As mentioned before, the confidence of a prediction from a *calibrated* classifier should reflect the probability that its prediction is correct. For instance, given 100 predictions with confidence of 0.8 (or 0.2), we expect that 80 (or 20) of them should be correctly classified. Mathematically, a well-calibrated classifier is defined as

$$\mathbb{P}(\hat{Y} = Y|\hat{P} = p) = p, \quad \forall p \in [0, 1], \tag{3.2}$$

where $\hat{Y}$ is the class prediction and $\hat{P}$ is its associated confidence (*e.g.*, the maximum probability obtaining by Softmax). It is shown in [131] that standard discriminative classifiers trained with cross-entropy loss are prone to be overconfident, meaning these models are less calibrated. Later work [132] empirically shows that models tend to be overconfident for samples with lower proximity [5] and under-confident for the samples with higher proximity. To mitigate the issue of miscalibration, there are roughly two types of methods consisting of 1) post-hoc methods and 2) training loss modification. Post-hoc methods can be further divided into *scaling-based* methods that include temperature scaling (TS), parameterized temperature scaling [133], and ensemble temperature scaling (ETS) [134], as well as *binning-based* methods such as classic histogram binning [135], mutual information maximization-based binning [136], and isotonic regression [137]. A taxonomy comparison of post-hoc calibration methods can be found in Table 3.3. Another line of work requires training with an additional loss such as [138], which yields better calibration via penalizing low-entropy output distributions. Recently, it is also empirically shown that regularization methods such as Mixup [62] and label smoothing [24] for improving classification accuracy also provide better calibrated predictions. Paper A falls into the category of modifying training loss via adding log-evidence loss during the training of classifiers.

Expected calibration error (ECE) is a commonly utilized metric to evaluate the miscalculation of a model. It is done by first grouping the samples into M bins according to their predicted confidence, and then setting the height

[5] Low proximity data (*i.e.*, data lying in the sparse region of the data distribution [132].

Methods	Equation	Parameters
Matrix scaling [131]	$\hat{\mathbf{p}} = \text{Softmax}(\mathbf{W}\mathbf{z} + \mathbf{b})$	$\mathbf{W}, \mathbf{b}$
Vector scaling [131]	$\hat{\mathbf{p}} = \text{Softmax}(\mathbf{W}\mathbf{z} + \mathbf{b})$	$\mathbf{W} = \text{diag}(w_1, w_2, \ldots, w_n), \mathbf{b}$
Temperature scaling (TS) [131]	$\hat{\mathbf{p}} = \text{Softmax}(\mathbf{z}/T)$	T
Parametrized temperature scaling (PTS) [133]	$\hat{\mathbf{p}} = \text{Softmax}(\mathbf{z}/T(\mathbf{z};\theta))$	T is sample-dependent.
PTS with k-NN [132]	$\hat{\mathbf{p}} = \text{Softmax}(\mathbf{z}/(\alpha d(\mathbf{x}))), \quad d(\mathbf{x}) = \frac{1}{k}\sum_{i\in\mathcal{N}_k(\mathbf{x})}\lVert f(\mathbf{x}) - f(\mathbf{x}_j)\rVert$	α, k
Ensemble temperature scaling (ETS) [134]	$\hat{\mathbf{p}} = w_1\,\text{Softmax}(\mathbf{z}) + w_2\,\text{Softmax}(\mathbf{z}/T) + w_3\frac{1}{C}$	T, w_1, w_2, w_3
Histogram binning [135]	$\hat{p}_{\text{HB}} = \frac{1}{S_b}\sum_{i\in S_b} y_i$, y_i is the ground-truth label (0 or 1)	–
Isotonic regression [137]	$\min_{\theta,\mathbf{a}} \sum_{m=1}^{M}\sum_{i=1}^{n} \mathbb{1}(a_m \le \hat{p}_i < a_{m+1})(\theta_m - y_i)^2$, s.t. $a = a_1 \le a_2 \le \cdots \le a_{M+1} = 1, \theta_1 \le \theta_2 \le \cdots \le \theta_M$	$\theta, \mathbf{a}$
Mutual information maximization-based binning [136]	$\hat{p}_{\text{MIMB}} = p(y = 1\lvert\hat{p} \in B_i)$, $\mathbf{B}^* = \arg\max_{\mathbf{B}} I(\hat{p}; y)$	$\mathbf{B}$ is bin boundaries.

Table 3.3: *Taxonomy Comparison of Post-hoc Calibration Methods.* $\mathbf{x}$ is an input, $\mathbf{z}$ denotes logits, and $\hat{\mathbf{p}}$ denotes the calibrated prediction.

of the bins to the average precision of the contained samples. *i.e.*, for samples x_i with confidence $\hat{p}_i$, B_m are the samples with $\hat{p}_i \in I_m = (\frac{m-1}{M}, \frac{m}{M})$, and

$$\begin{aligned} \text{acc}(B_m) &= \tfrac{1}{|B_m|} \sum\nolimits_{i \in B_m} \mathbb{1}(\hat{y}_i = y_i) \\ \text{conf}(B_m) &= \tfrac{1}{|B_m|} \sum\nolimits_{i \in B_m} \hat{p}_i, \end{aligned} \tag{3.3}$$

where $\hat{y}_i$ is the predicted label of sample i and y_i is the corresponding ground truth. Intuitively, a perfectly calibrated classifier satisfies $\text{acc}(B_m) = \text{conf}(B_m)$ for all $m \in \{1, \cdots, M\}$. Therefore, ECE [139] is defined as the difference between $\text{acc}(B_m)$ and $\text{conf}(B_m)$ to quantify the miscalibration, which is

$$\text{ECE} \coloneqq \sum\nolimits_{m=1}^{M} \tfrac{|B_m|}{N} \big|\text{acc}(B_m) - \text{conf}(B_m)\big|, \tag{3.4}$$

where N is the total number of samples.

Calibration for regression

Similarly, given the probability 90%, a calibrated regressor should output the prediction interval that covers 90% of ground truths [140]. In regression, neural networks should output a cumulative distribution function(CDF) F_i targeting y_i. Assuming $F_i^{-1} : [0, 1] \to \mathcal{Y}$ denotes the quantile function $F_i^{-1}(p) = \inf\ \{y : p \leq F_i(y)\}$. Mathematically, a well-calibrated regressor is defined as

$$\frac{\sum_{i=1}^{N} \mathbb{I}\left\{y_i \leq F_i^{-1}(p)\right\}}{N} \to p \quad \text{for all} \quad p \in [0, 1] \tag{3.5}$$

as $N \to \infty$.

There are few approaches dedicated to calibrating neural networks for regression tasks [140]–[143]. In this thesis, Paper B aims to obtain a better-calibrated regressor via joint energy-based modeling. [141] relies on a held-out calibrated dataset to match the predicted CDF and empirical frequency resulting in a better calibrated regressor. Such quantile-level calibration does not ensure calibration for a specific prediction. For instance, a regressor might provide an estimated mean μ and standard deviation σ, for predictions. It do not necessarily imply that the distribution of the actual outcomes for these predictions follows a Gaussian distribution with moments (μ, σ). Therefore, distribution calibration [142] aims to obtain distribution-level calibration, which

offers more accurate confidence for a continuous target variable. To be specific, it utilizes Beta calibration maps to transform the predicted CDF of a regressor. The parameters (a, b, c) of a Beta calibration map are learned by a Gaussian Process. Maximum mean discrepancy (MMD) [140] achieves a calibrated regressor by minimizing the kernel embedding measure. The resulting loss functions include negative log likelihood loss and MMD distance loss. Formally, the sample version of MMD over two distributions $\mathcal{P}$ and $\mathcal{Q}$ is defined as follows:

$$\hat{L}_m^2(\mathcal{P}, \mathcal{Q}) = \left\| \frac{1}{N} \sum_{i=1}^{N} \phi(y_i) - \frac{1}{M} \sum_{j=1}^{M} \phi(\hat{y}_j) \right\|_{\mathcal{F}}^2, \tag{3.6}$$

where N and M are the number of ground-truth targets drawn from target distribution $\mathcal{P}$ and random samples from the predictive distribution $\mathcal{Q}$, respectively. $\phi(\cdot)$ is a mixture of k radial basis function (RBF) kernels, *i.e.*,

$$\phi(x) = k(x, x') = \sum_{i=1}^{K} k_{\sigma_i}(x, x'). \tag{3.7}$$

Calibration error [141] representing the difference between p_j and $\hat{p}_j$ is commonly utilized to quantify the miscalibration. The first step is to choose M confidence levels $0 \leq p_1 < p_2 < \cdots < p_M \leq 1$, and then compute the empirical frequency

$$\hat{p}_j = \frac{|\{y_i | F_i(y_i) \leq p_j, i = 1, \cdots, N\}|}{N}, \tag{3.8}$$

where $F_i(y_i)$ is the cumulative distribution function (CDF) and T is the number of samples in the dataset. A perfect calibrated regressor is expected to have $p_j = \hat{p}_j$ for all $j \in (1, \ldots, M)$. Consequently, calibration error is defined as

$$\text{cal}(F_1, y_1, \cdots, F_N, y_N) \coloneqq \sum\nolimits_{j=1}^{M} \beta_j \cdot (p_j - \hat{p}_j)^2, \tag{3.9}$$

where the scalars β_j are weights. A toy example is given below to demonstrate how to evaluate calibration error for regression tasks.

Quantile-based calibration evaluation For regression tasks, we assume the outputs of a model follow a Gaussian distribution parameterized by $\mu_\theta(\mathbf{x})$

and $\sigma_\theta(\mathbf{x})$. When calculating the calibration error, Z-score can be utilized to calculate the coverage range while fixing the expected confidence level. We firstly set the expected confidence level to p_j. Then, Z-table is utilized to find the corresponding Z-score α. The corresponding range is $[\mu_\theta(\mathbf{x}) - \alpha\sigma_\theta(\mathbf{x}), \mu_\theta(\mathbf{x}) + \alpha\sigma_\theta(\mathbf{x})]$. Finally, we count the number of samples N' whose ground truth labels fall into this coverage range. Intuitively, the empirical or observed confidence level is

$$\hat{p}_j = \frac{N'}{N}, \tag{3.10}$$

where N represents the number of test samples.

Related research problems

Calibration evaluation Apart from ECE, maximum calibration error (MCE) and the maximum mean calibration error (MMCE) [144] are also commonly utilized metrics for quantifying calibration error in classification tasks. MCE focuses on the bin with maximum calibration error highlighting the worst-case error. MMCE is an alternative to ECE that avoids binning using a kernel-based approach to estimate calibration error. However all these three metrics are biased estimators as discussed in [145] because of the binning size, the sample size in each bin, or the kernel bandwidth in the MMCE. Few works [132], [145] aim to devise better calibration estimators. For instance, [145] argues that the common notion of calibration utilized in [131] is weak because it only considers the prediction with the highest probability. A stronger notion is to consider the predictions from all classes based on a predictive distribution. While [132] observes that the confidence of predictions depends on the data proximity, *i.e.*, the model tends to output overconfident (underconfident) predictions for the samples with lower (higher) proximity such that the miscalibration errors are canceled out. Therefore, [132] proposes a variant of expected calibration error considering the data proximity bias.

Conformal prediction Unlike temperature scaling [131], that directly modifies the logit obtained from the network without compromising the accuracy. Another appealing framework to achieve model calibration is conformal prediction (CP), which operates on a theoretical basis and guarantees the marginal coverage in practice. CP is briefly introduced here to give a more

comprehensive perspective. For more information, please refer to [146], [147]. A detailed discussion about temperature scaling and CP can be found in [148]. Let $(X_i, Y_i) \sim P, i = 1, \cdots m$ be the i.i.d. data and label pairs, from a distribution P on $\mathcal{X} \times \mathcal{Y}$, and consider a dataset with three splits including $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{cal}}$, and $\mathcal{D}_{\text{val}}$. The goal of CP is to convert a "point predictor" to a "set predictor" with a predefined error rate α denoted by $\mathcal{C}_\alpha(X_{n+1})$, where n is the size of calibration set. As mentioned, CP guarantees the marginal coverage, meaning

$$\mathbb{P}(Y_{n+1} \in \mathcal{C}_\alpha(X_{n+1})) \geq 1 - \alpha. \tag{3.11}$$

However, the conditional coverage is not necessarily guaranteed, *i.e.*,

$$\mathbb{P}(Y_{n+1} \in \mathcal{C}_\alpha(X_{n+1}) | X_{n+1} = \mathbf{x}) \geq 1 - \alpha. \tag{3.12}$$

Typically, the calculation of the prediction set consists of the following steps for a given model f_θ parametrized by θ, trained on $\mathcal{D}_{\text{train}}$:

1. Define an uncertainty score function $S(\mathbf{x}, y) \in \mathbb{R}$ (often referred to as the non-conformity score function, *e.g.*, $S(\mathbf{x}, y) = |f_\theta(\mathbf{x}) - y|$);
2. Calculate the non-conformity scores $\{(s_i)\}_{i=1}^{n_{\text{cal}}}\}$ for each data in $\mathcal{D}_{\text{cal}}$;
3. Given a user-specified error rate α, compute $\hat{q}$ as the $\frac{\lceil (n_{\text{cal}}+1)(1-\alpha) \rceil}{n_{\text{cal}}}$ quantile of the non-conformity scores $\{(s_i)\}_{i=1}^{n_{\text{cal}}}$;
4. Derive the confidence intervals or sets for the validation set, *e.g.*, if an absolute error is selected as the non-conformal score, the resulting interval is $[f(\mathbf{x}_i) - \hat{q}, f(\mathbf{x}_i) + \hat{q}]$.

3.3 Hallucination mitigation

As we discussed earlier, LLMs and LVLMs (also known as foundation models), suffer from the issue of hallucination [149], [150]. Specifically, the hallucination in LLMs refers to the phenomena that they occasionally generate unfaithful, fabricated, inconsistent, or nonsensical content [13] while in VLMs refers to the scenario that they sometimes produce responses which are not grounded in the visual input [150]. In this section, a brief summary of hallucination in LLMs is presented first and followed by a detailed literature review regarding hallucination in VLMs. It is worthwhile to note that the architecture design of

VLMs commonly encapsulates a language decoder. Therefore, one paragraph regarding latent representation in language models is also included for a better understanding the decoding mechanism. In this thesis, Paper F focuses on mitigating object hallucination in VLMs.

Hallucination mitigation in LLMs

Hallucination in neural machine translation (NMT) is first observed and presented in [151], which empirically shows that NMT systems are prone to generating highly flawed translations that are entirely disconnected from the source content. The emergence of hallucination in LLMs occurs when transformer-based models such as GPT-2 [152] and Bidirectional Encoder Representations from Transformers (BERT) [153] were adopted in the community. With the release of GPT-3 [154] and its impressive generative capabilities, hallucination in LLMs has been a central topic of concern for researchers working on AI alignment and factual response generation. An instance of hallucination is shown in Figure 3.5. The type of hallucination can be roughly categorized into

- Intrinsic hallucination: the case that the generated responses deviate from the input of users or the content that is generated previously [13];
- Extrinsic hallucination: the scenario that the generated responses are not grounded in the factual world knowledge [13]. Retrieval augmentation generation (RAG) [155] is a common approach to mitigating such hallucination.

The existing benchmarks often evaluate the ability of either *generating* factual statements or *discriminating* them from the non-factual ones [149]. The former tasks (*i.e.*, open-ended generation) are difficult to evaluate by nature, which heavily relies on human experts following specific guidelines. The latter tasks are much easier to evaluate via calculating the accuracy and truthfulness.

Latent representations in language models Understanding the decoding mechanism of transformer-based language decoders can directly contribute to mitigating hallucination. It has been studied from various perspectives including but not limited to attention maps/patterns [156]–[159] and the intermediate representation [160]–[165] with the application of early exiting [163], [164]

or model knowledge editing [166], [167]. Model knowledge editing refers to identifying and removing a (linear) concept subspace from the representation, preventing any (linear) predictor from recovering the concept. Meanwhile, early exiting in the context LLMs refers to projecting the hidden states extracted at each layer to the learned "unembedding" matrix of the language decoder. By doing this, one can obtain multiple distributions for the subsequent decoding step.

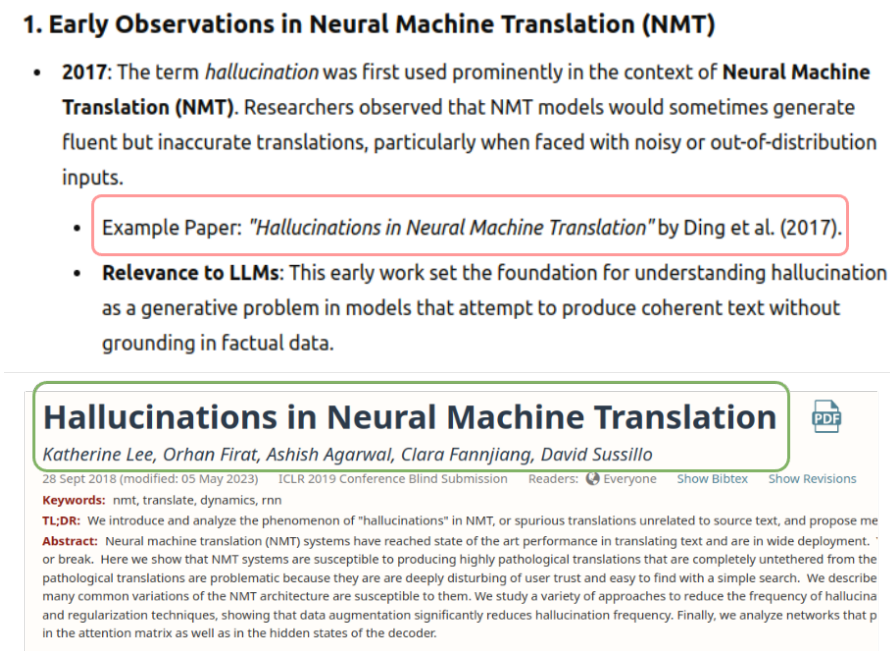

Figure 3.5: *Extrinsic Hallucination in GPT-4o.* The title of the paper is correct, but the names of the authors are incorrect. The incorrect part is highly in pink and the correct information is highlighted in green.

Hallucination mitigation in VLMs

Hallucination in VLMs refers to the scenario that they sometimes produce responses which are not grounded in the visual input. The problem itself can be traced back to [168], which is the initial work to investigate the issue of

object hallucination in image captioning tasks. Because of the potential applications of LVLMs, significant efforts have been dedicated to mitigate object hallucination since 2023. Several studies have focused on addressing this issue through: 1) fine-tuning LVLMs by replacing the CLIP encoder with the DINOv2 encoder [169]; 2) fine-tuning LVLMs using curated training data, where each sample pairs an image with a hallucinatory description, and the correct description serves as the output target [170]; 3) refining LVLMs by adding an extra head after the language decoder to predict visual tokens [171]; and 4) constructing revised token distributions for subsequent decoding [172]–[174]. The final types of methods are appealing because of their simplicity (*i.e.*, being training-free and compatible with different architectures). Meanwhile, a recent survey [150] categorizes object-related hallucinations into three groups: 1) a *category* group, where the VLM identifies incorrect or non-existing objects in the image; 2) an *attribute* group, where wrong description such as color and shape for the given visual input is generated; 3) a *relation* group, where incorrect relationship or interactions between objects are reported. In the following paragraph, we first go through the commonly-used benchmarks along with their evaluation metrics. Further, a detailed review of post-hoc methods (*i.e.*, training free) devised for mitigating hallucination in VLMs are summarized.

Datasets	Hallucination-types	# Pairs
MSCOCO [175]	category	9,000
A-OKVQA [176]	category	9,000
GQA [177]	category	9,000
MME [178]	category, attribute	240
MMVP [169]	category, attribute, relation	300

Table 3.4: *Specifications of Object Hallucination Benchmarks.*

Hallucination evaluation The existing benchmarks used to assess the extent of hallucination in VLMs can be roughly categorized into *discriminative tasks generative* tasks. The dataset summary is shown in Table 3.4.

The evaluation metrics include Caption Hallucination Assessment with Image Relevance (CHAIR) and GPT4-Assisted Visual Instruction Evaluation (GAVIE) [179] for generative tasks, as well as, accuracy, F1 score, and yes-

ratio gap for discriminative tasks. Two variants of CHAIR, *i.e.*, CHAIR_I for evaluating the degree of hallucination at the object instance level and CHAIR_S for evaluating at the sentence level. Mathematically,

$$\text{CHAIR}_I = \frac{|\{\text{hallucinated objects}\}|}{|\{\text{all mentioned objects}\}|}, \tag{3.13}$$

$$\text{CHAIR}_S = \frac{|\{\text{captions with hallucinated objects}\}|}{|\{\text{all captions}\}|}. \tag{3.14}$$

One notable pitfall of CHAIR_I is that it lacks contextual understanding while overemphasizes on individual instances. It is also reported in [180] that the calculation of CHAIR is sensitive to variations in instruction design, even when the semantic meaning remains similar. Another alternative evaluation metric is GAVIE, which consider to measure both accuracy (*i.e.*, whether the response is grounded in the visual input) and relevance (*i.e.*, whether the response directly follows the instruction) [179]. However, the evaluation process requires GPT-4 [181] to acts as the judge, which sometimes introduces unreliability to the evaluation procedure.

The evaluation metrics for discriminative tasks include accuracy, F1 score, and yes-ratio gap. A common experimental set-up for discriminative benchmarks such as POPE [180] is that each image is equipped with 6 questions, and half answers are "yes" and half answers are "no". Therefore, yes-ratio gap is to depict the gap between the predicted and the expected yes ratio, which reflects the degree of bias directly. To be specific, the yes ratio gap is defined as

$$\Delta_{\text{gap}} = \left| \frac{\#\text{ of answers with yes}}{\#\text{ of total questions}} - 0.5 \right|, \tag{3.15}$$

where $|\cdot|$ denotes the absolute value and 0.5 represents the expected yes ratio because the dataset is balanced. F-score (F1 score) is the harmonic mean of precision and recall. A more general F-score is denoted by F_β, where β is a positive real factor and is chosen such that recall is considered β times as important as precision. Formally,

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}, \tag{3.16}$$

where precision $= \frac{\text{TP}}{\text{TP+FP}}$ and recall $= \frac{\text{TP}}{\text{TP+FN}}$. The benchmarks such as POPE [180] choose $\beta = 1$ meaning precision and recall are equally important. Benchmarks such as THRONE [182] are concerned with measuring hallucination, thus they set $\beta = 0.5$ meaning precision is twice as important as recall.

Contrastive decoding in VLMs Contrastive decoding is first introduced as a method to reduce hallucination in LLMs [183]. Specifically, it leverages two LLMs with different capabilities (*i.e.*, one is the "expert" and the other is "amateur") and contrasts the predictive distribution from two LLMs, *i.e.*, $\log p_{\text{expert}(x_i<x_{<i})} - \log p_{\text{amateur}}(x_i < x_{<i})$. Subsequently, the resulting contrastive distribution is utilized for decoding. The motivation is that the amateur models tend to assign the highest confidence/probability to a repetitive token. By contrasting, such undesired behaviors occurred in amateur models can be factored out. Similarly, DoLa [184] adopts a similar approach but without relying on external LLMs. It identifies that knowledge bias mainly arises from the early layers of a model. Therefore, they propose to mitigate the factual hallucination by contrasting the predictive distributions from different layers within the same LLM. Naturally, a similar principle can also be applied to VLMs to mitigate object hallucination [172], [174], [185], [186]. VCD [172] observes that perturbed images with additive Gaussian noise are more prone to hallucination, where the predicted outputs are largely influenced by a language prior. To address this, the final logits are calculated as a weighted combination of those generated from the original and perturbed images. VDD [185] follows a similar principle to VCD, but adds a calibration step. Specifically, it learns a weight matrix W to adjust the predictive distribution from a noisy image, transforming it into a uniform distribution for each potential answer. The same weighted logit combination principle as VCD is then applied. Instruction contrastive decoding (ICD) [186] extends the contrastive principle to the introductions/prompts literally by adding a prefix (*e.g.*, `You are a confused object detector`) to the standard prompt to further amplify the hallucination. Similarly, the calculation of the final logit is the same as VCD [172] and VDD [185]. Most contrastive decoding methods for hallucination mitigation operate within internal states and require a contrasted distribution from either a distorted visual input [172], [185], or a predefined layer bucket [184], or prompt engineering [186]. It is also worthwhile to note that all contrastive-based methods necessitate an adaptive plausibility

constraint $\mathcal{H}_{\text{head}}$, which aims to restrict the effect of contrastive objective to the tokens in which the expert model is highly confident. Formally,

$$\mathcal{H}_{\text{head}}(x_{<i}) = \{x_i \in \mathcal{H}_{\text{head}} : p_{\text{expert}}(x_i < x_{<i}) \geq \alpha \max_w p_{\text{expert}}(w|x_{<i})\}, \tag{3.17}$$

where α is a hyperparameter in $[0, 1]$ that truncates the next token distribution of p_{expert}. Intuitively, larger α results in a more aggressive truncation meaning only the tokens with high probabilities are preserved and vice versa.

Non-contrastive decoding in VLMs Several works aim to mitigate hallucination without relying on contrasting another next token distribution [173], [187]. CGD [187] aims to mitigate object hallucination at a sentence level. Particularly, it leverages the powerful vision-language alignment capabilities of CLIP to identify sentences that are better aligned with the corresponding visual embeddings. This ensures that the generated responses not only have higher sentence likelihood but also higher CLIP scores. That is to say, they are less hallucinatory. However, its performance gain highly relies on the capability of external models. Further, the possible decoding methods are redistricted to nucleus sampling [36] and beam search in order to create candidate sentences. OPERA [173] stands out for its uniqueness of not requiring any "contrastive" logits. It observes the phenomenon that the presence of hallucination correlates with certain "knowledge aggregation patterns", *i.e.*, VLMs tend to generate new tokens by focusing on a few summary tokens but not necessarily taking all the previous tokens into account. Therefore, the hallucination is mitigated by penalizing the "over-trust" logit. However, the hysteresis of beam search necessitates a mechanism named retrospection-allocation, *i.e.*, the decoding procedure may roll back to the identified summary token and select other candidates for the next token prediction except for the candidates selected before. Consequently, OPERA [173] iteratively operates with the beam search decoding, which results in high-computational demand at the inference stage but also severely restricts its applicable scenarios. Our method proposed in Paper F is highly efficient, which only requires one single forward pass to calculate the energy score at each layer.

Methods	Free of					
	pre-defined layers	visual editing	prompt tuning	specific decoding	external knowledge	contrastive decoding
ICD [186]	✓	✓	✗	✓	✓	✗
DoLa [184]	✗	✓	✓	✓	✓	✗
CGD [187]	✓	✓	✓	✓	✗	✓
VCD [172]	✓	✗	✓	✓	✓	✗
OPERA [173]	✓	✓	✓	✗	✓	✓
HALC [174]	✗	✓	✓	✓	✗	✗
Energy-guided (Ours)	✓	✓	✓	✓	✓	✓

Table 3.5: *Taxonomy of Object Hallucination Mitigation Methods.*

Related research problems

Despite constructing a better predictive distribution in a post-hoc manner is practically appealing, mitigating hallucinations during pre-training or fine-tuning offers a more comprehensive solution to the problem. Therefore, the community aims to improve the multi-modal alignment of VLMs to address the issue of hallucination through the lens of vision encoder [31], [169], [188], language decoder [171], and different types of connectors between visual tokens and text tokens. MM1 [189] also looks into how different types of training data (*i.e.*, image + text data, interleaved data, synthetic data, and text only data) contribute to specific downstream tasks. For instance, image-caption data is beneficial for zero-shot tasks, and text-interleaved and text-only data are useful for few-shot and text-only tasks. Taking the architecture of LLaVA [30] as an example, several pioneering works aiming to improve feature alignment of VLMs are briefly discussed below from the perspective of vision-encoder and language decoder.

Visual encoder pre-training/fine-tuning One potential reason of hallucination in VLMs is the limited capability of the visual encoder. That is to say, the visual encoder might not be powerful enough to encode all the information contained in an image, resulting in defective visual tokens. Consequently, the language decoder cannot fully perceive the image, which can lead to hallucinatory responses. Commonly, the default visual encoder utilized in VLMs is CLIP [25], which is trained in a contrastive manner. Instead, AIM [188] proposes to train the visual encoder in an autoregressive way. Specifically, assuming an input image is split into patches without an overlapping region, the learning objective is to force the model to predict the next patch in raster

order. Recently, MMVP [169] empirically reveals that the feature embeddings of two visually distinct images, as generated by CLIP, have a smaller distance (measured by cosine similarity) compared to those generated by DINOv2. Additionally, [190] systematically examines how various visual tokenizers contribute to the performance of VLMs. However, the employed architecture is similar to Flamingo [191], which necessitates a perceiver resampler.

Language decoder fine-tuning A common design of language decoder employed in VLMs is generating language responses for a given prompt. MetaMorph [171] instead challenges this design and adds an extra head to the language decoder aiming to predict the visual tokens, which is learned by maximizing the cosine similarity between the original visual tokens and the predictive ones. Furthermore, it is empirically shown that the predictive visual tokens processed by stable diffusion could generate better images compared to the original visual tokens (*i.e.*, image embeddings extracted from CLIP).

CHAPTER 4

Summary of included papers

This chapter provides a summary of the included papers.

4.1 Paper A

Xixi Liu, D Staudt, Che-Tsung Lin, Christopher Zach
Effortless Training of Joint Energy-Based Models with Sliced Score Matching
International Conference on Pattern Recognition (ICPR)
pp. 2643-2649, 2022
©DOI: 10.1109/ICPR56361.2022.9956495 .

JEM [82] argues that standard discriminative classifiers can be upgraded to *joint energy-based models* (JEMs) by combining the classification loss with a log-evidence loss. Hence, such models intrinsically allow detection of out-of-distribution (OOD) samples, and empirically also provide better calibrated posteriors, *i.e.*, prediction uncertainties. However, the training procedure suggested for JEMs (using stochastic gradient Langevin dynamics—or SGLD—to maximize the evidence) is reported to be brittle. In this work we propose to

utilize score matching—in particular sliced score matching—to obtain a stable training method for JEMs. We observe empirically that the combination of score matching with the standard classification loss leads to improved OOD detection and better calibrated classifiers for otherwise identical DNN architectures. Additionally, we also analyze the impact of replacing the regular soft-max layer for classification with a gated soft-max one in order to improve the intrinsic transformation invariance and generalization ability.

4.2 Paper B

Xixi Liu, Che-Tsung Lin, Christopher Zach
Energy-based Models for Deep Probabilistic Regression
International Conference on Pattern Recognition (ICPR)
pp. 2643-2649, 2022
©DOI: 10.1109/ICPR56361.2022.9956495 .

Inspired by recent joint energy-based models for classification, in this work, we propose to utilize joint energy modeling for regression tasks. Within this framework, we apply our method to three computer vision regression tasks. We demonstrate that joint energy-based models for deep probabilistic regression improve the calibration property, do not require expensive inference, and yield competitive accuracy in terms of the mean absolute error (MAE).

4.3 Paper C

Xixi Liu, Yaroslava Lochman, Christopher Zach
GEN: Pushing the limits of softmax-based out-of-distribution detection
IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)
pp. 23946-23955, 2023
©DOI: 10.1109/CVPR52729.2023.02293 .

Out-of-distribution (OOD) detection has been extensively studied in order to successfully deploy neural networks, in particular, for safety-critical applications. Moreover, performing OOD detection on large-scale datasets is closer to the reality, but is also more challenging. Several approaches need to

either access the training data for score design or expose models to outliers during training. Some post-hoc methods are able to avoid the aforementioned constraints, but are less competitive. In this work, we propose Generalized ENtropy score (GEN), a simple but effective entropy-based score function, which can be applied to any pre-trained softmax-based classifier. Its performance is demonstrated on the large-scale ImageNet-1k OOD detection benchmark. It consistently improves the average AUROC across six commonly-used CNN-based and visual transformer classifiers over a number of state-of-the-art post-hoc methods. The average AUROC improvement is at least 3.5%. Furthermore, we use GEN on top of feature-based enhancing methods as well as methods using training statistics to further improve the OOD detection performance. The code is available at: https://github.com/XixiLiu95/GEN.

4.4 Paper D

Xixi Liu, Jennifer Alvén, Ida Häggström, Christopher Zach
Deep Nearest Neighbors for Anomaly Detection in Chest X-Rays
International Workshop on Machine Learning in Medical Imaging (MIML), held in conjunction with MICCAI
pp. 293–302, 2023
©DOI: 10.1007/978-3-031-45676-3-30 .

Identifying medically abnormal images is crucial to the diagnosis procedure in medical imaging. Due to the scarcity of annotated abnormal images, most reconstruction-based approaches for anomaly detection are trained only with normal images. At test time, images with large reconstruction errors are declared abnormal. In this work, we propose a novel feature-based method for anomaly detection in chest x-rays in a setting where only normal images are provided during training. The model consists of lightweight adaptor and predictor networks on top of a pre-trained feature extractor. The parameters of the pre-trained feature extractor are frozen, and training only involves fine-tuning the proposed adaptor and predictor layers using Siamese representation learning. During inference, multiple augmentations are applied to the test image, and our proposed anomaly score is simply the geometric mean of the k-nearest neighbor distances between the augmented test image features and the training image features. Our method achieves state-of-the-art results on two challenging benchmark datasets, the RSNA Pneumonia Detec-

tion Challenge dataset, and the VinBigData Chest X-ray Abnormalities Detection dataset. Furthermore, we empirically show that our method is robust to different amounts of anomalies among the normal images in the training dataset. The code is available at: https://github.com/XixiLiu95/deep-kNN-anomaly-detection.

4.5 Paper E

Xixi Liu, Christopher Zach
TAG: Text Prompt Augmentation for Zero-Shot Out-of-Distribution Detection
Published in European Conference on Computer Vision (ECCV)
pp. 364-380, 2024
©DOI: 10.1007/978-3-031-73464-9-22 .

Out-of-distribution (OOD) detection has been extensively studied for the reliable deployment of deep-learning models. Despite great progress in this research direction, most works focus on discriminative classifiers and perform OOD detection based on single-modal representations that consist of either visual or textual features. Moreover, they rely on training with in-distribution (ID) data. The emergence of vision-language models allows to perform zero-shot OOD detection by leveraging multi-modal feature embeddings and therefore only rely on labels defining ID data. Several approaches have been devised but these either need a given OOD label set, which might deviate from real OOD data, or fine-tune CLIP, which potentially has to be done for different ID datasets. In this paper, we first adapt various OOD scores developed for discriminative classifiers to CLIP. Further, we propose an enhanced method named *TAG* based on Text prompt AuGmentation to amplify the separation between ID and OOD data, which is simple but effective, and can be applied on various score functions. Its performance is demonstrated on CIFAR-100 and large-scale ImageNet-1k OOD detection benchmarks. It consistently improves AUROC and FPR95 on CIFAR-100 across four commonly used architectures over four baseline OOD scores. The average AUROC and FPR95 improvements are 6.35% and 10.67%, respectively. The results for ImageNet-1k follow a similar, but less pronounced pattern. The code is available at: `https://github.com/XixiLiu95/TAG`.

4.6 Paper F

Xixi Liu, Ailin Deng, Christopher Zach
Energy-Guided Decoding for Object Hallucination Mitigation
Submitted for Review, 2025 .

To ensure the reliable deployment of large vision language models (LVLMs) in the real world, particularly for safety-critical applications, it is essential to resolve the issue of hallucination, *i.e.* LVLMs occasionally generating contents that are not grounded in the visual inputs. Existing methods either demand sophisticated modifications to visual inputs [172], are restricted to specific decoding strategies [173], or rely on knowledge from other models [187]. In this work, we identify a significant imbalance in the yes ratio, *i.e.* the fraction of "yes" answers among the total number of questions, within VLMs. In order to mitigate this hallucinatory behavior we propose an energy-based decoding method, which dynamically select the hidden states from the layer with minimal energy score. It is simple and effective in reducing the bias for the yes ratio and boosting performance across three discriminative benchmarks (POPE [180], MME [178], and MMVP [169]). Our method consistently improves accuracy and F1 score on POPE benchmark across two commonly used VLMs over three baseline methods. The average accuracy improvement is 4.37% compared to the greedy decoding. Moreover, the proposed method is less biased in terms of yes ratio.

CHAPTER 5

Concluding Remarks and Future Work

This thesis is dedicated to enhancing the reliability of deep models, focusing on three critical applications: out-of-distribution (OOD) detection, model calibration, and hallucination mitigation. Paper A and Paper D mainly focus on addressing OOD detection and anomaly detection at the training stage. Specifically, Paper A focuses on devising an additional loss function for standard discriminative classifiers within the framework of joint energy-based modeling (JEM). Paper D aims to design an efficient framework with access to only normal images for detecting medical anomalies in Chest X-rays. The benefit is that it requires neither medical anomalies nor reconstructing normal images. Paper B focuses on extending the framework of JEM from classification to regression tasks, resulting in a better calibrated regressor while achieving competitive performance across three computer vision tasks. Paper C and Paper E tackle the problem of large-scale OOD detection without fine-tuning. Specifically, Paper C proposes an entropy-based OOD score that only accesses the probability information while achieving superior performance. It enhances the reliability of deep models in constrained scenarios. Paper E utilizes the powerful image-text alignment existing in contrastive vision-language models (VLMs) and enables zero-shot OOD detection (*i.e.*, without

accessing in-distribution (ID) images). It first adapts various OOD scoring methods, originally devised for discriminative classifiers, to contrastive VLMs (*i.e.*, CLIP [25]). Furthermore, an enhanced method named TAG, based on Text prompt AuGmentation, is proposed to amplify the separation between ID and OOD data, which is simple yet effective, and can be applied to various scoring methods. Paper F leverages the insights gained from OOD detection to resolve the issue of object hallucination existing in generative VLMs. It proposes an energy-guided decoding method that seeks to identify the layer with the minimum energy, where the output hidden states are projected onto the vocabulary head for subsequent decoding.

Through a comprehensive exploration of these topics, this thesis not only highlights the importance of reliable deep models but also provides practical algorithms and frameworks to achieve this goal, contributing to trustworthy AI systems for societal benefits. Moreover, this thesis also summarizes existing research and key findings related to these three applications. Hopefully, this comprehensive review will be useful for newcomers interested in this field.

5.1 Future work

In this section, I will mainly present my thoughts and insights on contributing to trustworthy AI, focusing on the challenges and limitations in the current research problems as well as potential research questions.

OOD detection

When we talk about OOD detection, we specifically refers to detecting samples with semantic shift compared to the training samples. This problem itself is a well-defined research question with standard benchmarks for evaluation. Currently, the state-of-art performance on these benchmarks is close to saturation indicating that much effort has been put into this direction. Meanwhile, it also motivates researchers to reflect whether the current benchmark is far from the realistic settings. As suggested in [124], a more practical setting is to reject samples that are misclassified ID as well as semantic OOD samples. Therefore, a natural research question to consider is as follows.

RQ1: When there is a limited budget for rejection samples, it is valuable to investigate which OOD score works best in such cases and to devise new OOD score if the performance of existing ones is not satisfactory.

Meanwhile, detecting semantically OOD samples can be beneficial to other computer vision tasks, one potential use of the OOD score is to select ID samples in the context of open-set semi-supervised learning (OSSL) to facilitate the annotation of unlabeled data. It would be valuable to explore which OOD score is most effective/robust for this task.

CLIP has demonstrated strong performance in zero-shot OOD detection. However, for large-scale OOD detection, its performance still requires fine-tuning with a few ID samples to match the performance of a fully supervised setting [117]. While the text-prompt augmentation method proposed in Paper E enhances OOD detection, there is still a performance gap compared to the supervised setting. Therefore, it would be interesting to investigate why this performance improvement is limited and how to address this issue.

RQ2: Understand and reduce the modality gap existing in CLIP [25] through devising new losses.

Hallucination mitigation

As the importance of LLMs and VLMs (*e.g.*, ChatGPT and GPT-4(V)) continues to grow, addressing hallucination mitigation has become increasingly urgent. However, most existing methods either rely on the contrastive decoding principle or require specific decoding mechanisms.

RQ1: Develop a plug-and-play decoding method that offers greater flexibility across different models and decoding strategies.

The root cause of hallucination in VLMs is much more complicated than LLMs due to the architecture design. One potential reason is the defective alignment between visual embedding and language embedding, which has been investigated by [169]. However, the improvement is still limited. I suspect that

the parametric knowledge represented in the weights of the language decoder is greatly biased. Therefore, how to inject visual knowledge into the language decoder is the key to resolve such issue. A pioneering work, MetaMorph [171], has explored this direction.

RQ2: How to enhance text-image alignment in generative VLMs via injecting visual knowledge into the language decoder?

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[2] OpenAI, *ChatGPT*, `https://openai.com/blog/chatgpt/`, 2023.

[3] OpenAI, *GPT-4V(ison) System Card*, `https://cdn.openai.com/papers/GPTV_System_Card.pdf`, 2023.

[4] *Be My Eyes: Bringing Sight to Blind and Low-Vision People*, `https://www.bemyeyes.com`.

[5] Microsoft, *Seeing AI - Microsoft Garage*, `https://www.seeingai.com`.

[6] A.-M. Marcu, L. Chen, J. Hünermann, *et al.*, "Lingoqa: Visual question answering for autonomous driving," *arXiv:2312.14115*, 2023.

[7] C. Li, C. Wong, S. Zhang, *et al.*, "Llava-med: Training a large language-and-vision assistant for biomedicine in one day," *arXiv:2306.00890*, 2023.

[8] N. Wiener, "Some moral and technical consequences of automation," *Science*, vol. 131, pp. 1355–1358, 1960.

[9] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in ai safety," *arXiv: 1606.06565*, 2016.

[10] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan, "Do deep generative models know what they don't know?" In *International Conference on Learning Representations (ICLR)*, 2019.

[11] *PictureThis*, `https://www.picturethisai.com/`.

[12] OpenAI, *GPT-4 System Card*, `https://cdn.openai.com/papers/gpt-4-system-card.pdf`, 2023.

[13] L. Weng, *Extrinsic hallucinations in llms.* `https://lilianweng.github.io/posts/2024-07-07-hallucination`, 2020.

[14] O. Russakovsky, J. Deng, H. Su, *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, 2015.

[15] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[16] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "Repvgg: Making vgg-style convnets great again," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[17] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[18] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *British Machine Vision Conference (BMVC)*, 2016.

[19] A. Kolesnikov, L. Beyer, X. Zhai, *et al.*, "Big transfer (bit): General visual representation learning," in *European Conference on Computer Vision (ECCV)*, 2020.

[20] Z. Liu, Y. Lin, Y. Cao, *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[21] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations (ICLR)*, 2021.

[22] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, "Training data-efficient image transformers & distillation through attention," in *International Conference on Machine Learning (ICML)*, 2021.

[23] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[24] R. Müller, S. Kornblith, and G. Hinton, "When does label smoothing help?" In *Advances in Neural Information Processing Systems*, 2020.

[25] A. Radford, J. W. Kim, C. Hallacy, *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning (ICML)*, 2021.

[26] J. Zhang, J. Huang, S. Jin, and S. Lu, "Vision-language models for vision tasks: A survey," *arXiv:2304.00685*, 2024.

[27] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," in *Advances in Neural Information Processing Systems*, 2019.

[28] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," *arXiv:2303.15343*, 2023.

[29] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, "Minigpt-4: Enhancing vision-language understanding with advanced large language models," *arXiv:2304.10592*, 2023.

[30] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *arXiv:2304.08485*, 2023.

[31] H. Liu, C. Li, Y. Li, and Y. J. Lee, "Improved baselines with visual instruction tuning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 26 296–26 306.

[32] W. Dai, J. Li, D. Li, *et al.*, "InstructBLIP: Towards general-purpose vision-language models with instruction tuning," in *Advances in Neural Information Processing Systems*, 2023.

[33] J. Li, D. Li, S. Savarese, and S. Hoi, "BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models," in *International Conference on Machine Learning (ICML)*, 2023.

[34] W.-L. Chiang, Z. Li, Z. Lin, *et al.*, *Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality*, `https://lmsys.org/blog/2023-03-30-vicuna/`, 2023.

[35] A. Ghosh, A. Acharya, S. Saha, V. Jain, and A. Chadha, "Exploring the frontier of vision-language models: A survey of current methodologies and future directions," *arXiv:2404.07214*, 2024.

[36] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," *arXiv:1904.09751*, 2020.

[37] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv:1211.3711*, 2012.

[38] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International Conference on Machine Learning (ICML)*, 2020.

[39] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Computer Vision and Pattern Recognition Conference (IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR))*, 2020.

[40] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," *arXiv:2111.06377*, 2021.

[41] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *European Conference on Computer Vision*, 2018.

[42] X. Chen and K. He, "Exploring simple siamese representation learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[43] M. Caron, H. Touvron, I. Misra, *et al.*, "Emerging properties in self-supervised vision transformers," in *the International Conference on Computer Vision (ICCV)*, 2021.

[44] M. Oquab, T. Darcet, T. Moutakanni, *et al.*, *Dinov2: Learning robust visual features without supervision*, 2023.

[45] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in *International Conference on Machine Learning (ICML)*, 2021.

[46] J.-B. Grill, F. Strub, F. Altché, *et al.*, "Bootstrap your own latent: A new approach to self-supervised learning," in *Advances in Neural Information Processing Systems*, 2020.

[47] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *Advances in Neural Information Processing Systems*, 2020.

[48] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[49] P. Khosla, P. Teterwak, C. Wang, *et al.*, "Supervised contrastive learning," in *NeuraIPS*, 2020.

[50] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" In *Advances in Neural Information Processing Systems*, 2017.

[51] J. Gawlikowski, C. R. N. Tassi, M. Ali, *et al.*, "A survey of uncertainty in deep neural networks," *Artificial Intelligence Review*, pp. 1513–1589, 2023.

[52] Y. Gal, "Uncertainty in deep learning," *Ph.D. Thesis*, 2016.

[53] M. Parry, A. P. Dawid, and S. Lauritzen, "Proper local scoring rules," *The Annals of Statistics*, 2012.

[54] A. Hyvärinen, "Estimation of non-normalized statistical models by score matching," *Journal of Machine Learning Research*, vol. 6, no. 4, pp. 695–709, 2005.

[55] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010.

[56] Y. Song, S. Garg, J. Shi, and S. Ermon, "Sliced score matching: A scalable approach to density and score estimation," *Uncertainty in Artificial Intelligence*, pp. 574–584, 2020.

[57] I. Krasin, T. Duerig, N. Alldrin, *et al.*, "Openimages: A public dataset for large-scale multi-label and multi-class image classification.," *Dataset available from https://github.com/openimages*, 2017.

[58] G. Van Horn, O. Mac Aodha, Y. Song, *et al.*, "The inaturalist species classification and detection dataset," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[59] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[60] H. Wang, Z. Li, L. Feng, and W. Zhang, "Vim: Out-of-distribution with virtual-logit matching," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[61] J. Yang, K. Zhou, Y. Li, and Z. Liu, "Generalized out-of-distribution detection: A survey," *arXiv:2110.11334*, 2021.

[62] S. Thulasidasan, G. Chennupati, J. Bilmes, T. Bhattacharya, and S. Michalak, "On mixup training: Improved calibration and predictive uncertainty for deep neural networks," in *Advances in Neural Information Processing Systems*, 2020.

[63] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[64] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *Advances in Neural Information Processing Systems*, 2018.

[65] W. Liu, X. Wang, J. Owens, and Y. Li, "Energy-based out-of-distribution detection," in *Advances in Neural Information Processing Systems*, 2020.

[66] D. Hendrycks, S. Basart, M. Mazeika, *et al.*, "Scaling out-of-distribution detection for real-world settings," in *International Conference on Machine Learning (ICML)*, 2022.

[67] R. Huang, A. Geng, and Y. Li, "On the importance of gradients for detecting distributional shifts in the wild," in *Advances in Neural Information Processing Systems*, 2021.

[68] K. Bibas, M. Feder, and T. Hassner, "Single layer predictive normalized maximum likelihood for out-of-distribution detection," in *Advances in Neural Information Processing Systems*, 2021.

[69] M. Cook, A. Zare, and P. Gader, "Outlier detection through null space analysis of neural networks," *arXiv:2007.01263*, 2020.

[70] X. Liu, Y. Lochman, and C. Zach, "Gen: Pushing the limits of softmax-based out-of-distribution detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[71] J. Park, Y. G. Jung, and A. B. J. Teoh, "Nearest neighbor guidance for out-of-distribution detection," in *IEEE/CVF International Conference on Computer Vision*, 2023, pp. 1686–1695.

[72] M. B. Ammar, N. Belkhir, S. Popescu, A. Manzanera, and G. Franchi, "NECO: NEural collapse based out-of-distribution detection," in *The Twelfth International Conference on Learning Representations*, 2024.

[73] L. Liu and Y. Qin, "Fast decision boundary based out-of-distribution detector," *International Conference on Machine Learning (ICML)*, 2024.

[74] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *International Conference on Learning Representations (ICLR)*, 2018.

[75] Y.-C. Hsu, Y. Shen, H. Jin, and Z. Kira, "Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[76] Y. Sun, C. Guo, and Y. Li, "React: Out-of-distribution detection with rectified activations," in *Advances in Neural Information Processing Systems*, 2021.

[77] Y. Song, N. Sebe, and W. Wang, "Rankfeat: Rank-1 feature removal for out-of-distribution detection," in *Advances in Neural Information Processing Systems*, 2022.

[78] A. Djurisic, N. Bozanic, A. Ashok, and R. Liu, "Extremely simple activation shaping for out-of-distribution detection," in *The Eleventh International Conference on Learning Representations*, 2023.

[79] K. Xu, R. Chen, G. Franchi, and A. Yao, "Scaling for training time and post-hoc out-of-distribution detection enhancement," in *International Conference on Learning Representations (ICLR)*, 2024.

[80] A. Djurisic, N. Bozanic, A. Ashok, and R. Liu, "Extremely simple activation shaping for out-of-distribution detection," in *International Conference on Learning Representations (ICLR)*, 2023.

[81] T. DeVries and G. W. Taylor, "Learning confidence for out-of-distribution detection in neural networks," *arXiv:1802.04865*, 2018.

[82] W. Grathwohl, K.-C. Wang, J.-H. Jacobsen, D. Duvenaud, M. Norouzi, and K. Swersky, "Your classifier is secretly an energy based model and you should treat it like one," in *International Conference on Learning Representations (ICLR)*, 2020.

[83] X. Liu, D. Staudt, C.-T. Lin, and C. Zach, "Effortless training of joint energy-based models with sliced score matching," in *2022 26th International Conference on Pattern Recognition (ICPR)*, 2022.

[84] R. Huang and Y. Li, "Mos: Towards scaling out-of-distribution detection for large semantic space," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[85] A. Zaeemzadeh, N. Bisagno, Z. Sambugaro, N. Conci, N. Rahnavard, and M. Shah, "Out-of-distribution detection using union of 1-dimensional subspaces," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[86] K. Lee, H. Lee, K. Lee, and J. Shin, "Training confidence-calibrated classifiers for detecting out-of-distribution samples," in *International Conference on Learning Representations (ICLR)*, 2018.

[87] E. Techapanurak, M. Suganuma, and T. Okatani, "Hyperparameter-free out-of-distribution detection using softmax of scaled cosine similarity," in *Asian Conference on Computer Vision (ACCV)*, 2019.

[88] H. Wei, R. Xie, H. Cheng, L. Feng, B. An, and Y. Li, "Mitigating neural network overconfidence with logit normalization," in *International Conference on Machine Learning (ICML)*, 2022.

[89] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," in *International Conference on Learning Representations (ICLR)*, 2019.

[90] A.-A. Papadopoulos, M. R. Rajati, N. Shaikh, and J. Wang, "Outlier exposure with confidence control for out-of-distribution detection," *Neurocomputing*, vol. 441, pp. 138–150, 2021.

[91] L. Tao, X. Du, J. Zhu, and Y. Li, "Non-parametric outlier synthesis," in *The Eleventh International Conference on Learning Representations*, 2023.

[92] X. Du, Z. Wang, M. Cai, and S. Li, "Vos: Learning what you don't know by virtual outlier synthesis," in *International Conference on Learning Representations (ICLR)*, 2022.

[93] X. Du, Y. Sun, X. Zhu, and Y. Li, "Dream the impossible: Outlier imagination with diffusion models," in *Advances in Neural Information Processing Systems*, 2023.

[94] G. A. Miller, "Wordnet: A lexical database for english," *Commun. ACM*, pp. 39–41, 1995.

[95] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[96] Y. Mao, F.-F. Xue, R. Wang, J. Zhang, W.-S. Zheng, and H. Liu, "Abnormality detection in chest x-ray images using uncertainty prediction autoencoders," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2020.

[97] D. Gong, L. Liu, V. Le, *et al.*, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[98] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurth, "F-anogan: Fast unsupervised anomaly detection with generative adversarial networks," *Medical Image Analysis*, 2019.

[99] J. Wolleb, F. Bieder, R. Sandkühler, and P. C. Cattin, "Diffusion models for medical anomaly detection," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2022.

[100] J. Wyatt, A. Leach, S. M. Schmon, and C. G. Willcocks, "Anoddpm: Anomaly detection with denoising diffusion probabilistic models using simplex noise," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2022.

[101] T. Denouden, R. Salay, K. Czarnecki, V. Abdelzad, B. Phan, and S. Vernekar, "Improving reconstruction autoencoder out-of-distribution detection with mahalanobis distance," *arXiv:1812.02765*, 2018.

[102] Y. Yang, R. Gao, and Q. Xu, "Out-of-distribution detection with semantic mismatch under masking," in *European Conference on Computer Vision (ECCV)*, 2022.

[103] Y. Zhou, "Rethinking reconstruction autoencoder-based out-of-distribution detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[104] B. Zong, Q. Song, M. R. Min, *et al.*, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *International Conference on Learning Representations (ICLR)*, 2018.

[105] L. Bergman, N. Cohen, and Y. Hoshen, "Deep nearest neighbor anomaly detection," in *arXiv*, 2020.

[106] Y. Sun, Y. Ming, X. Zhu, and Y. Li, "Out-of-distribution detection with deep nearest neighbors," in *International Conference on Machine Learning (ICML)*, 2022.

[107] Y. Ming, Y. Sun, O. Dia, and Y. Li, "Cider: Exploiting hyperspherical embeddings for out-of-distribution detection," *arXiv:2203.04450*, 2022.

[108] A. Ahmadian, Y. Ding, G. Eilertsen, and F. Lindsten, "Unsupervised novelty detection in pretrained representation space with locally adapted likelihood ratio," in *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, 2024, pp. 874–882.

[109] J. Ren, P. J. Liu, E. Fertig, *et al.*, "Likelihood ratios for out-of-distribution detection," in *Advances in Neural Information Processing Systems*, 2019.

[110] X. Liu, J. Alvén, I. Häggström, and C. Zach, "Deep nearest neighbors for unsupervised anomaly detection in chest x-ray images," 2023.

[111] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan, "Do deep generative models know what they don't know?" In *International Conference on Learning Representations*, 2019.

[112] P. Kirichenko, P. Izmailov, and A. G. Wilson, "Why normalizing flows fail to detect out-of-distribution data," in *Advances in Neural Information Processing Systems*, 2020.

[113] S. Fort, J. Ren, and B. Lakshminarayanan, "Exploring the limits of out-of-distribution detection," in *NeurIPS*, 2021.

[114] S. Esmaeilpourcharandabi, B. Liu, E. Robertson, and L. Shu, "Zero-shot open set detection by extending clip," in *AAAI*, 2021.

[115] Y. Ming, Z. Cai, J. Gu, Y. Sun, W. Li, and Y. Li, "Delving into out-of-distribution detection with vision-language representations," in *Advances in Neural Information Processing Systems*, 2022.

[116] H. Wang, Y. Li, H. Yao, and X. Li, "Clipn for zero-shot ood detection: Teaching clip to say no," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[117] A. Miyai, Q. Yu, G. Irie, and K. Aizawa, "Locoop: Few-shot out-of-distribution detection via prompt learning," in *Advances in Neural Information Processing Systems*, 2023.

[118] X. Jiang, F. Liu, Z. Fang, *et al.*, "Negative label guided OOD detection with pretrained vision-language models," in *The Twelfth International Conference on Learning Representations*, 2024.

[119] J. Yang, H. Wang, L. Feng, *et al.*, "Semantically coherent out-of-distribution detection," in *International Conference on Computer Vision*, 2021.

[120] F. Lu, K. Zhu, W. Zhai, K. Zheng, and Y. Cao, "Uncertainty-aware optimal transport for semantically coherent out-of-distribution detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[121] Y. Ming, H. Yin, and Y. Li, "On the impact of spurious correlation for out-of-distribution detection," *the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 10 051–10 059, 2022.

[122] Y. Chen, X. Zhu, W. Li, and S. Gong, "Semi-supervised learning under class distribution mismatch," *the AAAI Conference on Artificial Intelligence*, pp. 3569–3576, 2020.

[123] E. Wallin, L. Svensson, F. Kahl, and L. Hammarstrand, "Improving open-set semi-supervised learning with self-supervision," in *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024, pp. 2345–2354.

[124] G. Xia and C.-S. Bouganis, "Augmenting softmax information for selective classification with out-of-distribution data," in *Asian Conference on Computer Vision (ACCV)*, 2022.

[125] J. Kim, J. Koo, and S. Hwang, "A unified benchmark for the unknown detection capability of deep neural networks," *Expert Systems with Applications*, p. 120 461, 2023, ISSN: 0957-4174.

[126] J. Yang, K. Zhou, and Z. Liu, "Full-spectrum out-of-distribution detection," *arXiv:2204.05306*, 2022.

[127] D. J. C. MacKay, "A practical bayesian framework for backpropagation networks," *Neural Computation*, vol. 4, no. 3, pp. 448–472, 1992.

[128] J. S. Denker and Y. LeCun, "Transforming neural-net output levels to probability distributions," in *the 3rd International Conference on Neural Information Processing Systems (NeurIPS)*, 1990, pp. 853–859.

[129] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *International Conference on Machine Learning*, vol. 48, PMLR, 2016, pp. 1050–1059.

[130] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems*, 2017.

[131] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International Conference on Machine Learning (ICML)*, 2017.

[132] M. Xiong, A. Deng, P. W. Koh, *et al.*, "Proximity-informed calibration for deep neural networks," in *Advances in Neural Information Processing Systems*, 2023.

[133] C. Tomani, D. Cremers, and F. Buettner, "Parameterized temperature scaling for boosting the expressive power in post-hoc uncertainty calibration," in *European Conference on Computer Vision (ECCV)*, 2022.

[134] J. Zhang, B. Kailkhura, and T. Y.-J. Han, "Mix-n-match : Ensemble and compositional methods for uncertainty calibration in deep learning," in *International Conference on Machine Learning*, 2020.

[135] B. Zadrozny and C. Elkan, "Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers," in *International Conference on Machine Learning (ICML)*, 2001.

[136] K. Patel, W. H. Beluch, B. Yang, M. Pfeiffer, and D. Zhang, "Multi-class uncertainty calibration via mutual information maximization-based binning," in *International Conference on Learning Representations*, 2021.

[137] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates," in *ACM SIGKDD*, 2002.

[138] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton, "Regularizing neural networks by penalizing confident output distributions," in *International Conference on Learning Representations (ICLR)*, 2017.

[139] M. Pakdaman Naeini, G. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using bayesian binning," in *AAAI*, 2015.

[140] P. Cui, W. Hu, and J. Zhu, "Calibrated reliable regression using maximum mean discrepancy," in *Advances in Neural Information Processing Systems*, vol. 33, Curran Associates, Inc., 2020, pp. 17 164–17 175.

[141] V. Kuleshov, N. Fenner, and S. Ermon, "Accurate uncertainties for deep learning using calibrated regression," in *International Conference on Machine Learning (ICML)*, 2018, pp. 2801–2809.

[142] H. Song, T. Diethe, M. Kull, and P. Flach, "Distribution calibration for regression," in *the 36th International Conference on Machine Learning*, ser. Machine Learning Research, PMLR, 2019, pp. 5897–5906.

[143] X. Liu, C.-T. Lin, and C. Zach, "Energy-based models for deep probabilistic regression," in *2022 26th International Conference on Pattern Recognition (ICPR)*, 2022.

[144] A. Kumar, S. Sarawagi, and U. Jain, "Trainable calibration measures for neural networks from kernel mean embeddings," in *International Conference on Machine Learning (ICML)*, 2018.

[145] D. Widmann, F. Lindsten, and D. Zachariah, "Calibration tests in multi-class classification: A unifying framework," in *International Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[146] M. Sousa, "Inductive conformal prediction: A straightforward introduction with examples in python," *arXiv:2206.11810*, 2022.

[147] R. Tibshirani, "Conformal prediction," *UC Berkeley*, 2023.

[148] L. Dabah and T. Tirer, "On temperature scaling and conformal prediction of deep classifiers," *arXiv:2402.05806*, 2025.

[149] Z. et al., "Siren's song in the ai ocean: A survey on hallucination in large language models," *arXiv:2309.01219*, 2023.

[150] B. et al., "Hallucination of multimodal large language models: A survey," *arXiv:2404.18930*, 2024.

[151] A. Agarwal, C. Wong-Fannjiang, D. Sussillo, K. Lee, and O. Firat, "Hallucinations in neural machine translation," in *International Conference on Learning Representations (ICLR)*, 2018.

[152] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[153] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics, 2019, pp. 4171–4186.

[154] T. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2020, pp. 1877–1901.

[155] P. Lewis, E. Perez, A. Piktus, *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," in *Advances in Neural Information Processing Systems*, 2020.

[156] G. Kobayashi, T. Kuribayashi, S. Yokoi, and K. Inui, "Attention is not only a weight: Analyzing transformers with vector norms," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, 2020.

[157] A. Conmy, A. N. Mavor-Parker, A. Lynch, S. Heimersheim, and A. Garriga-Alonso, "Towards automated circuit discovery for mechanistic interpretability," in *Advances in Neural Information Processing Systems*, 2023.

[158] H. Chefer, S. Gur, and L. Wolf, "Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 397–406.

[159] K. Meng, D. Bau, A. Andonian, and Y. Belinkov, "Locating and editing factual associations in GPT," *Advances in Neural Information Processing Systems*, 2022.

[160] M. Geva, R. Schuster, J. Berant, and O. Levy, "Transformer feed-forward layers are key-value memories," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

[161] M. Geva, A. Caciularu, K. Wang, and Y. Goldberg, "Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space," in *Conference on Empirical Methods in Natural Language Processing*, 2022.

[162] G. Dar, M. Geva, A. Gupta, and J. Berant, "Analyzing transformers in embedding space," in *Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023.

[163] nostalgebraist, "Interpreting gpt: The logit lens.," 2024.

[164] D. Halawi, J.-S. Denain, and J. Steinhardt, "Overthinking the truth: Understanding how language models process false demonstrations," *arXiv:2307.09476*, 2023.

[165] N. Belrose, Z. Furman, L. Smith, *et al.*, "Eliciting latent predictions from transformers with the tuned lens," *arXiv:2303.08112*, 2023.

[166] T. Bolukbasi, K.-W. Chang, J. Zou, V. Saligrama, and A. Kalai, "Man is to computer programmer as woman is to homemaker? debiasing word embeddings," in *Advances in Neural Information Processing Systems*, 2016.

[167] S. Ravfogel, M. Twiton, Y. Goldberg, and R. Cotterell, "Linear adversarial concept erasure," *arXiv:2201.12091*, 2024.

[168] A. Rohrbach, L. Hendricks, K. Burns, T. Darrell, and K. Saenko, "Object hallucination in image captioning," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[169] S. Tong, Z. Liu, Y. Zhai, Y. Ma, Y. LeCun, and S. Xie, "Eyes wide shut? exploring the visual shortcomings of multimodal llms," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[170] Y. Zhou, C. Cui, J. Yoon, *et al.*, "Analyzing and mitigating object hallucination in large vision-language models," in *International Conference on Learning Representations (ICLR)*, 2024.

[171] S. Tong, D. Fan, J. Zhu, *et al.*, "Metamorph: Multimodal understanding and generation via instruction tuning," *arXiv:2412.14164*, 2024.

[172] S. Leng, H. Zhang, G. Chen, *et al.*, "Mitigating object hallucinations in large vision-language models through visual contrastive decoding," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[173] Q. Huang, X. Dong, P. zhang, *et al.*, "Opera: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation," *arXiv:2311.17911*, 2023.

[174] Z. Chen, Z. Zhao, H. Luo, H. Yao, B. Li, and J. Zhou, "Halc: Object hallucination reduction via adaptive focal-contrast decoding," *International Conference on Machine Learning (ICML)*, 2024.

[175] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, "Microsoft coco: Common objects in context," *arXiv:1405.0312*, 2015.

[176] D. Schwenk, A. Khandelwal, C. Clark, K. Marino, and R. Mottaghi, "A-okvqa: A benchmark for visual question answering using world knowledge," *arXiv:2206.01718*, 2022.

[177] D. A. Hudson and C. D. Manning, "Gqa: A new dataset for real-world visual reasoning and compositional question answering," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR*, 2019, pp. 6700–6709.

[178] C. Fu, P. Chen, Y. Shen, *et al.*, "Mme: A comprehensive evaluation benchmark for multimodal large language models," *arXiv:2306.13394*, 2023.

[179] F. Liu, K. Lin, L. Li, J. Wang, Y. Yacoob, and L. Wang, "Aligning large multi-modal model with robust instruction tuning," *arXiv:2306.14565*, 2023.

[180] Y. Li, Y. Du, K. Zhou, J. Wang, W. X. Zhao, and J.-R. Wen, "Evaluating object hallucination in large vision-language models," in *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[181] OpenAI, "Gpt-4 technical report," *arXiv:2303.08774*, 2023.

[182] P. Kaul, Z. Li, H. Yang, *et al.*, "Throne: An object-based hallucination benchmark for the free-form generations of large vision-language models," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[183] X. L. Li, A. Holtzman, D. Fried, *et al.*, "Contrastive decoding: Open-ended text generation as optimization," in *the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2023.

[184] Y.-S. Chuang, Y. Xie, H. Luo, Y. Kim, J. Glass, and P. He, "Dola: Decoding by contrasting layers improves factuality in large language models," *International Conference on Learning Representations (ICLR)*, 2024.

[185] Y.-F. Zhang, W. Yu, Q. Wen, *et al.*, "Debiasing large visual language models," *arXiv:2403.05262*, 2024.

[186] X. Wang, J. Pan, L. Ding, and C. Biemann, "Mitigating hallucinations in large vision-language models with instruction contrastive decoding," *arXiv:2403.18715*, 2024.

[187] A. Deng, Z. Chen, and B. Hooi, "Seeing is believing: Mitigating hallucination in large vision-language models via clip-guided decoding," *arXiv:2402.15300*, 2024.

[188] A. El-Nouby, M. Klein, S. Zhai, *et al.*, "Scalable pre-training of large autoregressive image models," in *the 41st International Conference on Machine Learning*, 2024, pp. 12 371–12 384.

[189] B. McKinzie, Z. Gan, J.-P. F. Biard, *et al.*, "Mm1: Methods, analysis insights from multimodal llm pre-training," in *European Conference on Computer Vision (ECCV)*, 2024.

[190] G. Wang, Y. Ge, X. Ding, M. Kankanhalli, and Y. Shan, "What makes for good visual tokenizers for large language models?" *arXiv: 2305.12223*, 2023.

[191] J.-B. Alayrac, J. Donahue, P. Luc, *et al.*, "Flamingo: A visual language model for few-shot learning," in *Advances in Neural Information Processing Systems*, 2022.

Part II

Papers

PAPER A

Effortless Training of Joint Energy-Based Models with Sliced Score Matching

Xixi Liu, D Staudt, Che-Tsung Lin, Christopher Zach

International Conference on Pattern Recognition (ICPR)
pp. 2643-2649, 2022
DOI: 10.1109/ICPR56361.2022.9956495

The layout has been revised.

Abstract

[a]Standard discriminative classifiers can be upgraded to *joint energy-based models* (JEMs) by combining the classification loss with a log-evidence loss. Hence, such models intrinsically allow detection of out-of-distribution (OOD) samples, and empirically also provide better calibrated posteriors, i.e. prediction uncertainties. However, the training procedure suggested for JEMs (using stochastic gradient Langevin dynamics—or SGLD—to maximize the evidence) is reported to be brittle. In this work we propose to utilize score matching—in particular sliced score matching—to obtain a stable training method for JEMs. We observe empirically that the combination of score matching with the standard classification loss leads to improved OOD detection and better calibrated classifiers for otherwise identical DNN architectures. Additionally, we also analyze the impact of replacing the regular soft-max layer for classification with a gated soft-max one in order to improve the intrinsic transformation invariance and generalization ability.

[a]This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation, and the Chalmers AI Research Center (CHAIR).

1 Introduction

Classification and regression tasks are the most successful application areas for deep learning. Nevertheless, desirable properties of any machine learning-based prediction methods are (i) the ability to indicate out-of-distribution (OOD) anomalies and (ii) to provide meaningful prediction confidences. OOD detection is key in real-world and safety-critical applications of machine learning since after deployment of a machine learning-based model the received inputs can be highly diverse and may therefore severely affect its behavior and performance. One situation that is especially important to avoid is that a model yields unreasonable but highly confident predictions for inputs obviously (to humans) not belonging to any of the trained classes (e.g., inputs

produced by DeepFool [1]). The ability to detect OOD samples is particularly essential for safety-critical applications such as rare disease identification and sensor failure detection.

A second requirement for a classifier to be eligible in a real-world decision-making system is that a valid assessment of the prediction confidence is provided. For instance, an autonomously driving car using a classifier to detect pedestrians and other objects should depend on other sensors if the confidence of a prediction based on camera input is low. Similarly, a system diagnosing mechanical faults displaying low confidence should be manually checked before resources are diverted. That is to say, it is often acceptable that a classifier is less accurate but instead well-calibrated, meaning that its prediction confidence is aligned with its miscalibration. More specifically, when the prediction confidence is 0.9, the classifier should have a 90% chance of being correct. Classifiers trained with standard classification losses, e.g., cross-entropy, tend to be over-confident (which is by design, as they are aiming for a perfect match between classification posterior and the pure ground-truth label distribution). Hence, trained classifiers are usually augmented in a post-processing step using a calibration method in order to calibrate the predicted posteriors. These calibration methods require the use of a hold-out validation dataset [2].

Interestingly, JEM [3] have demonstrated that combining a standard classification (cross-entropy) loss with the log-evidence from a probabilistic model yields neural networks that are (i) competitive in classification accuracy, (ii) deliver well-calibrated prediction confidences, and (iii) enable OOD detection. The key step is to interpret the logits of a classifier as joint log-likelihoods over inputs and target labels, leading to their proposed Joint Energy-based Model (JEM) approach. However, the resulting training method is admitted to be unstable and prone to divergence if the respective hyper-parameters are not tuned correctly. Successful trained of JEM requires multiple restarts from saved check-points with changed random seeds. As an alternative to the JEM training described in [3] we propose to combine score matching with the cross-entropy loss in order to stabilize the training process. Empirically, we obtain classification accuracy, OOD detection, and calibration results comparable to the ones reported for JEM, but with a absolutely straightforward and effortless training method.

2 Related work

Probabilistic modeling A probabilistic model $p(x;\theta)$ with parameters θ allows to evaluate the likelihood of a sample x and is,therefore, the basis of anomaly (or OOD) detection. Usually $p(\cdot;\theta)$ (or the corresponding log-likelihood $\log p(\cdot;\theta)$ is a general function e.g. represented by a neural network, hence direct estimation of the parameters θ from training data by maximizing the log-likelihood is intractable due to the lack of a closed-form partition function. Several sampling-based approximations to maximum likelihood training exist, such as contrastive divergence [4], [5] and Langevin dynamics [6]. Score matching [7] is an attractive alternative to sampling-based methods since it avoids the partition function entirely. Consequently, score matching yields unnormalized probabilities, which is sufficient to compare the likelihood of data samples. Score matching is a particular instance of a larger class of *local proper scoring rules* [8] (see [9], [10] for introductions to the general concept of *proper scoring rules*), and in certain settings it is strongly connected to denoising auto-encoders [11]. A recent extension of score matching is *sliced score matching* [12], which significantly improves the computational costs of score matching for high-dimensional input spaces and is, therefore, the basis for our approach.

OOD Detection Out-of-distribution (OOD) detection (and the closely related task of anomaly detection, see [13] for a recent survey) has recently received attention to make deep neural networks more robust in safety-critical application scenarios. In contrast to robust training (e.g. [14], OOD detection adds robustness at inference time. It can be implemented as an "add-on" to already trained networks [15]–[18] by using the classification posterior (sometimes referred as the softmax confidence) as the main guide. However, using the classifier posterior for OOD detection is problematic, since neural networks can assign high confidence to (specifically designed) OOD samples [1].

In the scenario where only in-distribution data is available, training a probabilistic model in order to model in-distribution samples is the classical approach. With a suitable model and training procedure, OOD samples are expected to have a low likelihood under the trained probabilistic model. Since expressive probabilistic models require highly non-linear regression networks, direct optimization of the maximum likelihood loss is unavailable, and extensions such as variational inference [19]–[21] or normalizing flows [22]–[24] are

typically employed. Using unnormalized probabilistic models (also known as *energy-based models*, EBMs) relaxes the normalization constraint of probability distributions and allows greater flexibility in the choice of the training loss (such as IGEBM—implicit generation with energy-based models [25]). In particular, auto-encoders are a prominent instance of EBMs used for OOD detection [26], [27]. Powerful probabilistic models solely trained from in-distribution data are not always the best tool to detect OOD samples as empirically verified in [28], which may be resolved by "regularization" of EBM training via a classification loss as considered in our work.

Calibration methods Most existing calibration methods are post-processing steps, requiring a hold-out validation set that can be the same as the one used for hyperparameter tuning. These methods can be divided into two types, depending on whether the model is binary or multinomial, and further subdivided into non-parametric and parametric methods[2]. For binary models, the non-parametric calibration methods include histogram binning [29] and isotonic regression [30]. The parametric approaches include Bayesian binning into quantiles(BBQ [31]) and Platt scaling [32]. Methods for multinomial models are extensions of those for binary models. Examples include matrix scaling, vector scaling, and temperature scaling, which are extensions of Platt scaling [2]. Matrix scaling applies a linear transformation $\boldsymbol{W}\boldsymbol{z_i} + \boldsymbol{b}$ to the logits. In the case of vector scaling, $\boldsymbol{W}$ is restricted to the diagonal. Temperature scaling is rather simple in that it only has a single scalar parameter T for all classes, as opposed to the two parameters of Platt scaling.

3 Background

3.1 Energy-based Models and JEM

Energy-based models (EBMs, e.g. [33]) are based on an energy function $E_\theta(\cdot)$ with parameters θ, that assigns an energy level $E_\theta(x)$ for each element x in an input space $\mathbb{R}^D$. An EBM induces an unnormalized probability distribution via $p_\theta(x) \propto \exp(-E_\theta(x))$ with associated (but usually intractable) partition function $Z(\theta) \coloneqq \int \exp(-E_\theta(x))\, dx$.

Joint energy-based models (JEMs [3]) use an EBM for the joint distribution $p_\theta(x, y)$, where (x, y) is a pair of input x and categorical class label y. After

observing that $\log p_\theta(x, y)$ can be written as

$$\log p_\theta(x, y) = \log p_\theta(x) + \log p_\theta(y|x), \tag{A.1}$$

where the first term is the log-evidence of x and the second term is the classification cross-entropy, it is suggested in [3] to use this decomposition to determine θ. In particular, the logit for given input x and class y, $f_\theta(x)[y]$, is re-interpreted as joint log-likelihood $\log p_\theta(x, y)$. The standard soft-argmax layer combined with the cross-entropy loss is exactly the 2nd term in Eq. (A.1),

$$\log p_\theta(y|x) = f_\theta(x)[y] - \text{Softmax}_{y'} f_\theta(x)[y']. \tag{A.2}$$

$\log p_\theta(x)$ can be obtained by marginalizing over y,

$$\begin{aligned} \log p_\theta(x) &= \log \sum\nolimits_y \log p_\theta(x, y) = \log \sum\nolimits_y f_\theta(x)[y] \\ &= \text{LogSumExp}_y f_\theta(x)[y]. \end{aligned} \tag{A.3}$$

Hence, $E_\theta(x) = -\log p_\theta(x)$ (and therefore the first term in Eq. A.1) has a closed-form expression. This term is optimized using a Monte-Carlo approximation and via stochastic gradient Langevin dynamics (SGLD) [6].

3.2 Score Matching

Score Matching (SM [7]) is a method to estimate unnormalized statistical models without explicit knowledge of the partition function. More specifically, the parameters of a model distribution are estimated by minimizing the squared distance between the *score functions* (i.e. the gradients of the log-density, $\nabla_x \log p(x)$) of the data and the model distribution, p_d and p_θ, respectively. The partition function of the model distribution does not appear in the objective (due to the derivative). Sliced score matching (SSM [12]) replaces the (vectorial) score function by respective projections onto random directions. In particular, we utilize the variance-reduced version of sliced score matching given by the following objective,

$$J_{\text{SSM}}(\theta) = \mathbb{E}_{\substack{x\sim p_d \\ v\sim p_v}} \left[\tfrac{1}{2} \|\nabla_x \log p_\theta(x)\|^2 + v^\top (\nabla^2 \log p_\theta(x)) v \right], \tag{A.4}$$

where ∇^2 refers to the Hessian, and p_v is a radially symmetric distribution to generate random directions. The finite sample version $\tilde{J}_{SSM}$ of J_{SSM} is obtained by averaging over the training set and by continuously sampling directions from p_v.

3.3 Reliability Diagram and Expected Calibrated Error

A reliability diagram is a way of visualizing if a classifier is well-calibrated or not [2]. It displays the relation between the confidence of the classifier and its actual accuracy. If the model is perfectly calibrated, the diagram will correspond to the identity function. This is done by first grouping samples into M bins according to their predicted confidence, and then setting the height of the bins to the average accuracy of the contained samples. I.e., for samples x_i with confidence $\hat{p}_i$, B_m are the samples with $\hat{p}_i \in I_m = (\frac{m-1}{M}, \frac{m}{M})$, and where $\hat{y}_i$ is the predicted label of sample i and y_i is the corresponding ground truth. A perfectly calibrated classifier satisfies $\text{acc}(B_m) = \text{conf}(B_m)$ for all $m \in \{1, \dots, M\}$.

However, the reliability diagram only serves as visualization. A scalar representing the difference between $\text{acc}(B_m)$ and $\text{conf}(B_m)$ is more convenient for quantitative comparison. We, therefore, use the Expected Calibration Error (ECE, [31]) to quantify the miscalibration. It is defined as

$$\text{ECE} \coloneqq \sum_{m=1}^{M} \frac{|B_m|}{N} \big|\text{acc}(B_m) - \text{conf}(B_m)\big|, \tag{A.5}$$

where N is the total number of samples.

3.4 Out-of-distribution Detection

Out-of-distribution detection can be considered as a binary classification problem, which aims to identify the samples that are different from the learned distribution. In other words, the model should be able to produce a score $s_\theta(\boldsymbol{x}) \in \mathbb{R}$ (not to be confused with the score function $\nabla_x \log p(x)$), where θ are the learned parameters, that represent the probability of x belonging to any known class. In this work, the (unnormalized) log-likelihood of the data point $\boldsymbol{x}$ is usually chosen to be the score $s_\theta(\boldsymbol{x})$. It is expected that in-distribution samples are assigned higher likelihoods (and therefore scores), and OOD samples are assigned lower likelihoods. The area under the receiver operating characteristics (AUROC) is used for evaluation. Note that calibration and OOD detection are orthogonal concepts: low prediction confidence can be caused by perfectly valid but otherwise hard to classify inputs.

4 Proposed Method

The main issue of JEM is the optimization of the log-evidence $\log p_\theta(x)$ via Stochastic Gradient Langevin Dynamics (SGLD [6]), which turns out to be unstable and prone to diverging behavior [3]. Therefore we propose to use score matching, in particular sliced score matching (SSM [12]), instead of SGLD. Consequently, our proposed *JEM-SSM* training loss reads as

$$\frac{1}{N} \sum_i \log p_\theta(y_i|x_i) + \lambda \tilde{J}_{\text{SSM}}(x_i), \tag{A.6}$$

where $\lambda > 0$ is a hyper-parameter. Since score matching only yields unnormalized probabilities, $\log p_\theta(x)$ is only estimated up to an unknown constant. Knowledge of unnormalized probabilities is sufficient for OOD detection. The objective in Eq. A.6 consists of two parts: first, it contains the cross-entropy classification loss, and the second term is the surrogate for the log-evidence. Under certain assumptions (in particular that the training data is sampled from $p_{\theta^*}(x, y)$, where θ^* are the true distribution parameters) the theory of proper scoring rules asserts consistency of the maximizer $\hat{\theta} \to \theta^*$ for $N \to \infty$. Similar to [3] we use the logits $f_\theta(x)[y]$ for the (now unnormalized) log-likelihoods $\log p^0_\theta(x, y)$, and the log-evidence is induced via $\log p^0_\theta(x) = \text{LogSumExp}_y f_\theta(x)[y]$ (see also Sec. 3.1). Since invariance to geometric image transformation is an increasingly important aspect of deep learning-based methods, we also investigate if substituting the vanilla soft-max layer with gated soft-max one (GSM, [34]) is beneficial in those settings. In this scenario, the input image in the original gated soft-max is replaced by CNN-provided feature maps. We employ the factorized variant of gated soft-max in two versions: the first variant, JEM-GSM$_1$, uses a 1x1 convolution to map the channels of those feature maps to a single channel, and the second variant, JEM-GSM$_{\text{multi}}$, treats the input to the gated soft-max layer as hyper-spectral image. Otherwise, we apply the same training method as with JEM-SSM.

4.1 A variation of the JEM objective

In this section we illustrate a modified JEM-SSM objective, which replaces the evidence $p_\theta(x)$ with a class-conditional model $p_\theta(x|y)$. One motivation for this variation is given by the fact that knowing the conditionals $p(y|x)$ and $p(x|y)$ is equivalent to knowledge of the joint distribution $p(x, y)$ [35]. Hence, it is in

principle sufficient to estimate the posteriors $p_\theta(y|x)$ and the class-conditional likelihoods $p_\theta(x|y)$. As it will be pointed out below, some attention needs to be paid since using score matching to estimate the parameters of $p_\theta(x|y)$ only allows identification of the corresponding unnormalized likelihood $q^0_\theta(x|y)$. We run experiments using this JEM variation (denoted by JEM-SSM$_v$) to assess whether conditional models are sufficient when marginalization w.r.t. target labels is not possible (such as in regression tasks).

We use the following model for the unnormalized joint distribution $p^0_\theta(x, y)$,

$$\log p^0_\theta(x, y) = f_\theta(x)[y] = (W g_\theta(x) + b)[y] = w_y^\top g_\theta(x) + b_y, \tag{A.7}$$

where the last layer (containing the logits) of the network f_θ is a linear layer with bias. w_y is the y-th row of W as column vector. Now $\log p^0_\theta(x|y)$ is given by

$$\log p^0_\theta(x|y) = w_y^\top g_\theta(x) + b_y - \log p(y), \tag{A.8}$$

but score matching will only yield

$$\log q^0_\theta(x|y) = w_y^\top g_\theta(x), \tag{A.9}$$

since $b_y - \log p(y)$ will vanish by taking the derivative w.r.t. x. Hence w_y and θ (the network parameters of g_θ common to all classes) appear in score matching, but not b_y. b_y is determined solely by the classification loss, and W and θ appear in both terms. After training b_y should approximate $\log Z - \log Z_y$. In JEM-SM, due to the marginalization over y, b_y appears in both terms of the loss function. At test time we evaluate either the (unnormalized) log-evidence $\log p^0_\theta(x)$ (JEM-SSM$_v$) or the class-conditional evidence $\log p^0_\theta(\mathbf{x})$ (JEM-SSM$^*_v$).

5 Experimental Results

We empirically investigate the performance of our proposed JEM-SSM method (and its variants such as JEM-GSM and JEM-SSM$_v$) in terms of classification accuracy, expected calibration error, and OOD detection ability. The baseline method is JEM and we use the results as reported in [3].

We generally use the same Wide Residual Network (WRN, [36]) architecture as used in the original JEM work [3] with the following general parameters:

the network depth is 28, the widening factor is 10, batch normalisation is disabled. For JEM-GSM, a gated soft-max layer with 11 hidden units and 40 filters is used in place of the regular soft-max classification layer. We differentiate JEM-GSM$_1$ and JEM-GSM$_{\text{multi}}$ as described in Section 3.1. In some additional experiments, we used a LeNet5 [37] model instead of the WRN in order to emphasize the potential impact of the GSM layer. The output of the LeNet5 convolutional backbone is directly connected with the GSM layer as in JEM-GSM$_1$.

Since JEM results were primarily reported on the CIFAR10 [38] dataset (with SVHN [39], CIFAR100, CIFAR10-Interp[1], and CelebA [40] as OOD datasets), many of our experiments also focus on these datasets. For the GSM experiments, we additionally evaluate smallNORB [41] and MNIST-Rot-12k[2], since these datasets explicitly contain images from multiple viewing directions and geometric transformations.All WRN models were trained for up to 150 epochs and LeNet5 models for up to 200. The hyper-parameter λ in Eq. (A.6) is fixed to 0.01 in all experiments.

5.1 Ease of Training

The primary advantage of using SSM over SGLD lies in it streamlining training procedures. [3] reported severe instabilities, preventing their model from converging unless it was restarted repeatedly from checkpoints. In contrast, all variations of JEM-SSM converged immediately and without any additional measures in our experiments. This persisted through all variations of hyper-parameters we performed. We hypothesize that this difference is due to the largely deterministic nature of the variance-reduced sliced score matching objective, as opposed to the high variance SGLD estimate. [3] considered these problems the biggest obstacle to widespread adaption of their model [3].

Table 2 shows the mean and standard deviation after training the models 50 times with different random seeds. Those experiments further demonstrate that we avoided the training problems of the original JEM.

[1] All images are interpolations between two images in CIFAR10.
[2] Rotated MNIST with 10k training samples, 2k validation, and 5k test

$s_\theta(\boldsymbol{x}) = \log p_\theta(x)$	SVHN ↑	CIFAR10-Interp ↑	CIFAR100 ↑	CelebA ↑	Acc. ↑	ECE ↓
Uncond. Glow [42]	0.05	0.51	0.55	0.57	67.60%	-
Class-Cond. Glow	0.07	0.45	0.51	0.53	-	-
IGEBM [25]	0.63	0.70	0.50	0.70	49.10%	-
JEM	0.67	0.65	0.67	0.75	**92.82%**	4.20%
JEM*	**0.83**	**0.78**	**0.82**	0.79	**92.82%**	4.20%
JEM-SSM	0.77	0.72	0.78	0.87	90.70%	2.62%
JEM-SSM$_v$	0.58	0.69	0.77	**0.89**	91.07%	**2.23%**
JEM-SSM$^*_v$	0.67	0.69	0.80	0.87	91.07%	**2.23%**
JEM-GSM$_1$	0.73	0.73	0.71	0.62	91.86%	3.00%
JEM-GSM$_{\text{multi}}$	0.40	0.69	0.74	0.67	91.25%	2.63%

Table 1: OOD detection results for models trained on CIFAR10. Values are AUROC. JEM* refers to using the score $s_\theta(\boldsymbol{x})$ proposed in [3] instead of $\log p_\theta(x)$. JEM-SSM$_v$ and JEM-SSM$^*_v$ are using the modified training objective (Sec. 4.1) with $s_\theta(x) = \log p^0_\theta(x)$ and $s_\theta(x) = \max_y \log p^0_\theta(x, y)$, respectively, used at test time.

5.2 Out of Distribution Detection

Table **??** shows our results for OOD detection with models trained on CIFAR-10 compared to several generative or hybrid models including JEM. The listed values are AUROC, using $s_\theta(x) = \log p_\theta(x)$ (or the unnormalized log-probability $\log p^0_\theta(x)$) as score function. Acc. refers to the accuracy achieved on the CIFAR-10 test set and indicates similar levels of accuracy for all JEM-derived models. The table also includes the results of training with the alternate objective proposed in section 4.1 as JEM-SSM$_v$.

$s_\theta(\boldsymbol{x}) = \log p_\theta(x)$	SVHN ↑	CIFAR100 ↑	CelebA ↑	Acc. ↑	ECE ↓
smallNORB$_{\text{shuffled}}$	0.57 ± 0.11	0.30 ± 0.07	0.09 ± 0.04	$99.74 \pm 0.42\%$	$0.11 \pm 0.26\%$
smallNORB$^*_{\text{shuffled}}$	$\mathbf{0.86 \pm 0.09}$	$\mathbf{0.89 \pm 0.06}$	$\mathbf{0.93 \pm 0.08}$	$99.60 \pm 0.12\%$	$3.30 \pm 0.53\%$
smallNORB$^{**}_{\text{shuffled}}$	0.50 ± 0.09	0.21 ± 0.05	0.04 ± 0.02	$\mathbf{99.80 \pm 0.17}\%$	$\mathbf{0.10 \pm 0.09}\%$
smallNORB$^{***}_{\text{shuffled}}$	0.79 ± 0.10	0.74 ± 0.09	0.62 ± 0.18	$99.72 \pm 0.15\%$	$1.70 \pm 0.47\%$
MNIST-rot-12k	0.82 ± 0.10	0.71 ± 0.10	0.65 ± 0.11	$89.78 \pm 0.35\%$	$5.70 \pm 0.56\%$
MNIST-rot-12k*	0.61 ± 0.06	0.66 ± 0.05	0.67 ± 0.05	$90.66 \pm 0.36\%$	$\mathbf{0.46 \pm 0.17}\%$
MNIST-rot-12k**	0.83 ± 0.16	0.75 ± 0.17	0.66 ± 0.20	$89.70 \pm 0.61\%$	$7.32 \pm 0.58\%$
MNIST-rot-12k***	$\mathbf{0.90 \pm 0.02}$	$\mathbf{0.87 \pm 0.02}$	$\mathbf{0.88 \pm 0.04}$	$\mathbf{91.05 \pm 0.45}\%$	$1.04 \pm 0.25\%$

Table 2: OOD detection results for LeNet5, LeNet5-SSM (indicated with *), LeNet5-GSM (**) and LeNet5-GSM-SSM (***), trained on the indicated data. Values are AUROC. Mean and standard deviation over 50 runs.

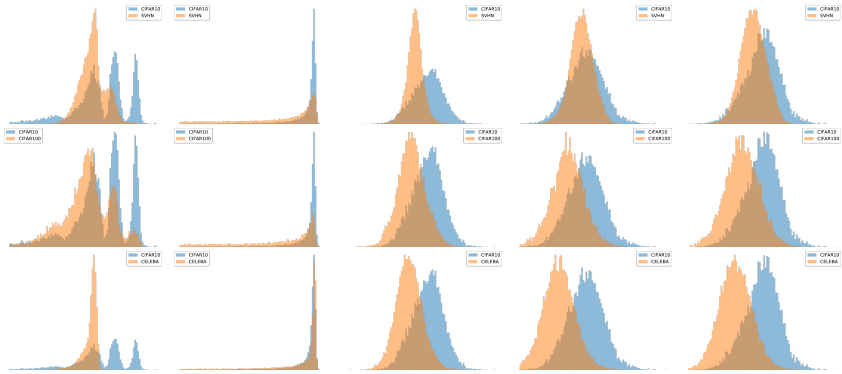

Figure 1: Histograms for OOD detection of models trained on CIFAR10. Blue shows the scores on the CIFAR10 dataset, orange on the sets listed in the first column.

All our proposed models show an improvement in the AUROC score compared to JEM at slightly reduced classification accuracy. JEM* refers to using the score $s_\theta(\boldsymbol{x})$ proposed in [3] instead of $\log p_\theta(x)$. In order to visualize these results, we plot histograms of the assigned values in Fig. 1. As can be seen, our models consistently yield higher scores for in-distribution samples. Interestingly, our models seem to form unimodal distributions in all cases, whereas JEM has learned three peaks for CIFAR-10.

We further examined what improvements training with SSM offered for weaker models on the example of LeNet5. Table 2 shows the mean and standard deviation of OOD detection and accuracies achieved when training it 50 times on MNIST-rot-12k and a shuffled version of smallNORB. The latter was done to put the focus on generalisation over transformations rather than the original intention of learning to recognise models that follow similar concepts (e.g., 'has four legs'). The small standard deviation of the evaluated quantities supports our claim of training stability for our JEM variations.

On smallNORB, both SSM and GSM-SSM improved OOD detection with only minor losses in accuracy. However, on MNIST-rot-12k both actually improved accuracy, but only GSM-SSM helped with OOD detection on all tested sets. This is likely due to the greater rotation variance offered by GSM.

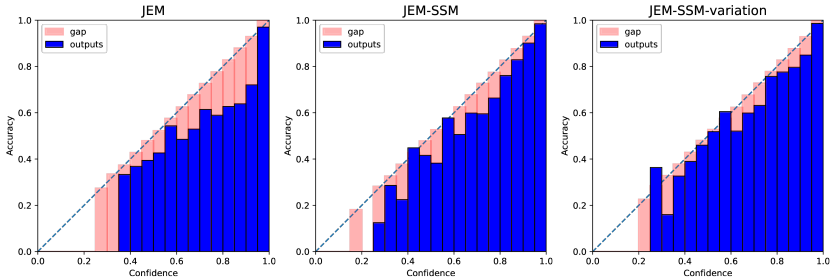

Figure 2: Reliability diagrams on CIFAR10 for JEM, JEM-SSM and JEM-SSM$_v$.

5.3 Calibration

To measure calibration, we used the Expected Calibration Error (ECE) described in section 3.3. The reliability diagrams described in the same section serve as visualisation.

As can be seen in the last column of Table **??**, all our models achieved a lower ECE than JEM, indicating that their accuracy more closely matches their predicted confidence values. This is further demonstrated in Fig. 2, where both JEM-SSM and JEM-SSM$_v$ more closely follow the diagonal, especially for higher confidence predictions. Outliers in the lower confidence areas can be explained by fewer samples falling into those bins. On CIFAR-100 this is even more extreme. The accuracy reduction is stronger, but JEM-GSM$_{\text{multi}}$ achieved a massive reduction in ECE (Table 3) that is also apparent in the associated reliability diagram in Fig. 4, showing a near-perfect match of confidence and accuracy.

Model	Accuracy%↑	ECE%↓
JEM	**72.20**	4.87
Baseline(Ours)	73.20	22.24
JEM-SSM	66.34	8.06
JEM-GSM$_1$	62.06	5.60
JEM-GSM$_{\text{multi}}$	64.62	**1.35**

Figure 3: Accuracy and ECE on CIFAR100.

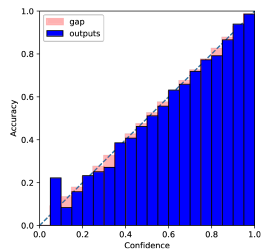

Figure 4: Reliability diagram on CIFAR100 for JEM-GSM.

Model	Accuracy (RGB) %↑	ECE (RGB) %↓	Accuracy (Grey) %↑	ECE (Grey) %↓
LeNet5	54.33	5.04	48.40	4.27
LeNet5-GSM	**70.59**	**1.25**	**68.55**	**1.62**
LeNet5-SSM	54.06	4.86	48.58	4.70
LeNet5-GSM-SSM	68.69	2.54	67.42	4.35

Table 3: Accuracy and ECE on CIFAR10 (RGB and grey-scale versions) with LeNet5.

5.4 Additional Gated Soft-Max Experiments

When using strong backbone networks, employing standard or gated soft-max yields minimal differences as seen in Table **??**. Therefore in this section, we additionally evaluate the standard and gated soft-max using weaker backbones, in particular ResNet18 and LeNet5. Since both LeNet5 and ResNet18 provide a single-channel feature map, the distinction between GSM_1 and $\text{GSM}_{\text{multi}}$ is not necessary for these experiments (unlike WideResNet used in the main paper).

For comparison with the JEM and JEM-GSM results, we first conduct experiments on CIFAR10. The results of this are shown in Table 3 (columns 2 and 3). Here, adding GSM provided a clear improvement in accuracy and ECE for both, training purely as a classifier and with additional SSM loss. These differences are even more strongly pronounced when the gray-scale variant of CIFAR10 is used as a dataset (Table 3, columns 4 and 5).

Since one of the primary advantages of GSM is the ability to leverage geometric invariances, we further conducted experiments on datasets with such transformations, in order to test if this property is retained in Deep GSM. For this, we used variations of CIFAR10 and MNIST that we call CIFAR10-Affine and MNIST-SomeRot. CIFAR10-Affine applies random, affine transformations on both, training and test CIFAR10 data, and MNIST-SomeRot denotes MNIST with all images randomly rotated in the test data, but only half of the digit classes being rotated in the training set (the other digits are provided solely as upright ones). Table 4 shows the accuracy and ECE scores achieved by LeNet5-based networks on MNIST-SomeRot, and Table 5 illustrates the results for CIFAR10-Affine using a ResNet-18 backbone. Replacing vanilla softmax with a gated softmax layer is clearly beneficial for improved accuracy, but comes at the cost of poorer calibrated classifiers.

Model	Accuracy%↑	ECE%↓
LeNet5	60.66	27.09
LeNet5-GSM	62.07	34.86
LeNet5-SSM	60.81	**22.39**
LeNet5-GSM-SSM	**63.13**	25.32

Table 4: Comparison of GSM on MNIST-SomeRot.

Model	Accuracy%↑	ECE%↓
ResNet18	58.66	1.73
ResNet18-GSM	**71.62**	4.42
ResNet18-SSM	57.83	**1.44**
ResNet18-GSM-SSM	70.94	3.29

Table 5: Comparison of GSM on CIFAR10-Affine.

5.5 Improving classifier calibration by temperature scaling

We applied temperature scaling [2] to improve the calibration performance of the classifiers. Table 6 depicts the ECE before and after temperature scaling. Post-processing the logits via temperature scaling leads to substantial improvements in all obtained ECE values. Training with a pure classification loss followed by temperature scaling is not sufficient to outperform models trained with a combined classification and SSM loss.

Model	$\text{ECE}_{\text{original}}\%\downarrow$	$\text{ECE}_{\text{ts}}\%\downarrow$
$\text{JEM}_{\text{class}}$	22.24	2.98
JEM-SSM	**4.28**	1.12
JEM-GSM_1	5.60	**0.93**

Table 6: ECE trained on CIFAR100 before and after applying temperature scaling. $\text{JEM}_{\text{class}}$ was trained with just classification loss and no SSM.

6 Conclusion

In many applications, a model being correct in its predictions is less important than knowing when it is likely incorrect. The ability of a model to supply confidence values close to its true level of certainty is referred to as calibration. Joint Energy-based Models (JEM) are models trained with a combined classification and log-evidence loss and have been demonstrated to be well-calibrated and to provide at the same time good out-of-distribution (OOD) detection capabilities. However, they have also proven to be highly unstable and therefore hard to train properly.

In this work, we propose to use sliced score matching instead of the log-

evidence loss, which results in substantially smoother and effortless training. Our models perform competitively with JEM in terms of OOD detection and classification accuracy and are often better calibrated. We additionally introduced an alternative objective for the log-evidence loss based on the joint distribution of input and class, improving calibration further.

References

[1] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[2] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International Conference on Machine Learning (ICML)*, 2017.

[3] W. Grathwohl, K.-C. Wang, J.-H. Jacobsen, D. Duvenaud, M. Norouzi, and K. Swersky, "Your classifier is secretly an energy based model and you should treat it like one," in *International Conference on Learning Representations (ICLR)*, 2020.

[4] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[5] M. A. Carreira-Perpinan and G. E. Hinton, "On contrastive divergence learning.," *AISTATS*, vol. 10, pp. 33–40, 2005.

[6] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient langevin dynamics," in *International Conference on Machine Learning (ICML)*, 2011.

[7] A. Hyvärinen, "Estimation of non-normalized statistical models by score matching," *Journal of Machine Learning Research*, vol. 6, no. 4, pp. 695–709, 2005.

[8] M. Parry, A. P. Dawid, and S. Lauritzen, "Proper local scoring rules," *The Annals of Statistics*, 2012.

[9] A. P. Dawid and M. Musio, "Theory and applications of proper scoring rules," *Metron*, 2014.

[10] T. Gneiting and A. E. Raftery, "Strictly proper scoring rules, prediction, and estimation," *Journal of the American statistical Association*, 2007.

[11] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural computation*, vol. 23, no. 7, pp. 1661–1674, 2011.

[12] Y. Song, S. Garg, J. Shi, and S. Ermon, "Sliced score matching: A scalable approach to density and score estimation," *Uncertainty in Artificial Intelligence*, pp. 574–584, 2020.

[13] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv:1901.03407*, 2019.

[14] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, "Learning from noisy labels with deep neural networks: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[15] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[16] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," *arXiv:1706.02690*, 2017.

[17] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *Advances in Neural Information Processing Systems*, 2018.

[18] Y.-C. Hsu, Y. Shen, H. Jin, and Z. Kira, "Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[19] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Machine learning*, vol. 37, no. 2, pp. 183–233, 1999.

[20] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv:1312.6114*, 2013.

[21] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[22] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *International Conference on Machine Learning (ICML)*, 2015.

[23] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," *arXiv:1605.08803*, 2016.

[24] E. Zisselman and A. Tamar, "Deep residual flow for out of distribution detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[25] Y. Du and I. Mordatch, "Implicit generation and modeling with energy based models," in *Advances in Neural Information Processing Systems*, 2019.

[26] T. Denouden, R. Salay, K. Czarnecki, V. Abdelzad, B. Phan, and S. Vernekar, "Improving reconstruction autoencoder out-of-distribution detection with mahalanobis distance," *arXiv:1812.02765*, 2018.

[27] Z. Xiao, Q. Yan, and Y. Amit, "Likelihood regret: An out-of-distribution detection score for variational auto-encoder," *arXiv:2003.02977*, 2020.

[28] E. T. Nalisnick, A. Matsukawa, Y. Teh, D. Görür, and B. Lakshminarayanan, "Do deep generative models know what they don't know?" *arXiv:1810.09136*, 2018.

[29] B. Zadrozny and C. Elkan, "Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers," in *International Conference on Machine Learning (ICML)*, 2001.

[30] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates," in *ACM SIGKDD*, 2002.

[31] M. Pakdaman Naeini, G. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using bayesian binning," in *AAAI*, 2015.

[32] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[33] Y. LeCun, S. Chopra, R. Hadsell, F. J. Huang, and et al., "A tutorial on energy-based learning," MIT Press, 2006.

[34] R. Memisevic, C. Zach, M. Pollefeys, and G. E. Hinton, "Gated softmax classification," in *Advances in Neural Information Processing Systems*, 2010.

[35] B. C. Arnold, E. Castillo, and J. M. Sarabia, "Conditionally specified distributions: An introduction (with comments and a rejoinder by the authors)," *Statistical Science*, vol. 16, no. 3, pp. 249–274, 2001.

[36] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *British Machine Vision Conference (BMVC)*, 2016.

[37] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[38] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[39] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Ng, "Reading digits in natural images with unsupervised feature learning," in *Advances in Neural Information Processing Systems*, 2011.

[40] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015.

[41] Y. LeCun, F. J. Huang, and L. Bottou, "Learning methods for generic object recognition with invariance to pose and lighting," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[42] E. Nalisnick, A. Matsukawa, Y. W. Teh, and B. Lakshminarayanan, "Detecting out-of-distribution inputs to deep generative models using typicality," in *Advances in Neural Information Processing Systems*, 2019.

PAPER B

Energy-based Models for Deep Probabilistic Regression

Xixi Liu, Che-Tsung Lin, Christopher Zach

International Conference on Pattern Recognition (ICPR)
pp. 2643-2649, 2022
DOI: 10.1109/ICPR56361.2022.9956495

The layout has been revised.

Abstract

It is desirable that a deep neural network trained on a regression task not only achieves high prediction accuracy, but its prediction posteriors are also well-calibrated, especially in safety-critical settings. Recently, energy-based models specifically to enrich regression posteriors have been proposed and achieve state-of-art results in object detection tasks. However, applying these models at prediction time is not straightforward as the resulting inference methods require to minimize an underlying energy function. Furthermore, these methods empirically do not provide accurate prediction uncertainties. Inspired by recent joint energy-based models for classification, in this work, we propose to utilize a joint energy model for regression tasks and describe architectural differences needed in this setting. Within this framework, we apply our methods to three computer vision regression tasks. We demonstrate that joint energy-based models for deep probabilistic regression improve the calibration property, do not require expensive inference, and yield competitive accuracy in terms of the mean absolute error (MAE).

1 Introduction

Regression is an important task in several computer vision and machine learning applications, including but not limited to object detection, head pose regression, and age estimation. Using deep neural network (DNN), the regression task is commonly done by learning a mapping $\phi(\cdot;\theta) : \mathbb{R}^d \mapsto \mathbb{R}^m$ from an input data point $x \in \mathbb{R}^d$ to an output target vector $y \in \mathbb{R}^m$. The model is then trained to find the optimal parameters θ^* that maximizes the overall likelihood based on a given training data set $\mathcal{D}$. While effective, directly predicting the target vector during testing, i.e., $\phi(x;\theta^*)$ only yields point prediction, while other important statistics of $p(x)$ are still missing. In recent years, it has been shown that uncertainty of the prediction is also crucial for determining the reliability when deploying a deep learning based model to safety-critical real world applications. For example, when the per-

ception system of an autonomous car are detecting objects in the perceived images, knowing how certain the object detector is on the prediction is essential. Besides, it is also expected to be well-calibrated. Therefore, modelling and quantifying the uncertainty of DNNs is a very active research field.

The uncertainty in DNNs are commonly divided in two types [1]. One is data (aleatoric) uncertainty, which is caused by the data and is irreducible. More specifically, the data uncertainty is caused by the information loss about input samples due to the error and noise in the measurement systems. The other is model (epistemic) uncertainty, which is caused by the model and is reducible. More precisely, the model uncertainty covers the uncertainty that is caused by the pitfalls in the model such as errors in the training procedure, an insufficient model structure, or lack of knowledge due to unknown samples or a bad coverage of the training data [2]. A well-calibrated regressor means the expected confidence level is aligned with its observed confidence level. e.g., a 80% posterior confidence interval is able to contain the true outcome 80% [3]. Quantitatively, calibration error [3], a non-negative score, is widely used for evaluate whether a regressor is well-calibrated or not. In general, calibration error is correlated with model uncertainty. If the model uncertainty could be entirely reduced, the data uncertainty could perfectly represent the real world information. Namely, we have a perfectly calibrated model.

Moreover, most of regression models are limited to model unimodal distribution, such as Gaussian or Laplace, which limits the expressiveness of learned models. In practice, the distribution of the data is more likely to be multimodal and complex. Clearly, such models are insufficient to fully represent the target density. Therefore, energy-based models (EBMs) are expected to resolve this issue because it can model the density of any observed data and does not require for proper normalization compared to most probabilistic models [4]. In general, EBMs are commonly used in generative modelling tasks [5]–[10]. Recently, several researchers explore to apply EBMs to regression task [11]–[13]. Empirically, it benefits several computer vision tasks including but not confined to object detection and visual tracking. However, none of current EBMs for regression address the issue of calibration. Especially, a well-calibrated regressor is quite essential when deploying deep learning-based models to safety-critical real-world applications. In short, calibration should take precedence over all other properties.

Contribution: In this paper, we proposed a different perspective on the

energy-based model for regression compared to the formulation proposed by [12]. Our method is built upon a standard regression model and able to model the joint energy $E(x, y)$. Our main contributions are:

1. We show that our model could achieve lower mean absolute error (MAE) as well as lower calibration error (CE) compared with several standard regression architectures.

2. We could achieve lower MAE without running gradient-based algorithms during the inference stage compared to [12].

3. We demonstrate our method on three different challenging computer vision tasks including object detection, age estimation, and head pose estimation compared to current state-of-art methods.

2 Background

Given a training dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$ containing N data points, where $x \in \mathbb{R}^d$ is the input data and $y \in \mathbb{R}^m$ is the desired target vector, deep regression is commonly done by training a deep neural network (DNN) ϕ_θ that aims to minimize the following L^2 loss:

$$\mathbb{E}_{(x,y)\sim\mathcal{D}} \|\phi_\theta(x) - y\|_2^2. \tag{B.1}$$

2.1 Predicting uncertainty using DNN

In the above model, we only have the point prediction. However, the model henceforth lacks information about the predicted uncertainty which is sometimes important for several tasks. Taking advantage of the learning capability of neural networks, one can also consider modelling the predictive distribution. For example, assuming the predictive distribution is Gaussian, the outputs would be mean $\phi_\theta(x)$ and covariance $\Sigma_\theta(x)$, which can be jointly predicted by the network, i.e.,

$$p(y|x;\theta) = \mathcal{N}(y; \phi_\theta(x), \Sigma_\theta(x)). \tag{B.2}$$

The corresponding loss function could be negative likelihood.

2.2 Mixture Density Networks and Deep Mixture Density Networks

The above discussion mostly concerns about uni-modal distributions such as Gaussian, while the data in practice can belong to more complicated distributions. Henceforth, to enrich the modelling capability, Mixture Density Networks (MDN [14]) is proposed and this work allows a better approximation of a real distribution of the data via combining a mixture model [15] and a conventional neural network.

In Mixture of Gaussian, the density $p(y|x)$ can be written as a mixture of K Gaussians,

$$p(y|x) = \sum_{k=1}^{K} w_k(x)\mathcal{N}(y; \phi_k(x), \Sigma_k(x)). \tag{B.3}$$

The parameters w_k are weights or mixing coefficients of each Gaussian component, which is defined by mean $\phi_k(x)$ and variance $\Sigma_k(x)$. Hence, Deep MDN is proposed to combine the strengths of DNN and MDN to train a neural network to predict $\{w_k, \phi_k, \Sigma_k\}_{k=1}^{K}$.

2.3 Energy-based models and score matching

Energy-based models (EBM) [16] aim to model unknown normalized distributions $p(x)$ by associating each data point to an energy function $E(x) : \mathbb{R}^d \mapsto \mathbb{R}$ corresponding to the negative logarithm of an un-normalized density function, i.e.,

$$p(x) = \frac{\exp(-E_\theta(x))}{Z(\theta)}, \tag{B.4}$$

with $Z(\theta) = \int_x \exp(-E_\theta(x))dx$. The difficulty in estimating the parameters of a general probabilistic model from data samples lies mainly in the requirement that valid probability distribution need to be normalized. The standard maximum-likelihood method requires estimation of the normalizing constant (or partition function), which can be obtained by Monte-Carlo methods. However, reasonably accurate estimates of the normalization constant is intractable in some cases, especially when the samples have high dimensionality (for continuous random variables) or large cardinality (for discrete

samples) [16]. Score matching (SM, [17]) is an alternative approach that bypasses the need for estimating the partition function entirely. In short, the parameters of a model distribution are estimated by minimizing the Fisher divergence between the data distribution and the model distribution [18]. More specifically, the squared distance between *score functions* (i.e. the gradients of the log-density, $\nabla_x \log p(x)$) of the data and the model distribution, p_d and p_θ, respectively, is minimized. Because the partition function does not occur in the score function, SM is often used to estimate the main parameters of unnormalized probabilistic models. The partition function required to obtain a normalized distribution can be estimated in a post-processing step if needed.

However, in practice SM is limited to rather low-dimensional problems due to the high computational cost for computing the Hessian of the model's log-density. This deficiency motivates several variants of SM including but not limited to denoising score matching (DSM [19]) and sliced score matching (SSM [18]). SSM substitutes the (vectorial) score function with respective projections onto random directions for saving the computation time. To stabilize the training procedure, we employ the variance-reduced version of SSM given by the following objective,

$$\begin{aligned} &J_{SSM}(\theta) = \\ &\mathbb{E}_{x\sim p_d, v\sim p_v}\left[\tfrac{1}{2}\|\nabla_x \log p_\theta(x)\|^2 + v^\top(\nabla^2 \log p_\theta(x))v\right], \end{aligned} \tag{B.5}$$

where ∇^2 is the 2nd derivative operator ($\nabla^2 := (\frac{\partial^2}{\partial x_i \partial x_j})_{i,j}$), and p_v is a radially symmetric distribution to generate random directions. The finite sample version $\tilde{J}_{SSM}$ of J_{SSM} is obtained by averaging over the training set and by continuously sampling directions from p_v.

2.4 Calibration Curve and Calibration Error for Regression

A predictive machine learning-based system is well-calibrated, if its provided prediction confidence matches the actual confidence (e.g. the prediction is correct for 80% of samples that have prediction confidence of at least 80%). Calibration methods can be generally grouped into three types according to the stage where they are applied [2]: (i) Regularization methods applied during the training phase, which is achieved via modifying the objective function or augmenting the training dataset. For example, [20] obtained better calibration via penalizing low entropy output distributions. (ii) Post-processing

methods applied after training the whole model. Such approaches [3] [21] require additional dataset to calibrate the pre-trained regressor without decreasing the accuracy. Similar to temperature scaling on classification tasks, [21] proposes to rescale the predicted standard deviation of a pre-trained network instead of logit in classification tasks. (iii) Methods that aim to reduce the model uncertainty to obtain a better calibrated regressor. [22] reduced the model uncertainty by learning parameters configurations and averaging over the resulting models.

A calibration curve is a way of visualizing whether a regression mapping is well-calibrated or not [3]. It displays the relation between the expected confidence level of the regressor and its observed confidence level. If the model is perfectly calibrated, the calibration curve will correspond to the identity function. To generate a calibration curve, the first step is to choose M confidence levels $0 \leq p_1 < p_2 < \cdots < p_M \leq 1$ and then compute the empirical frequency

$$\hat{p}_j = \frac{|\{y_t | F_t(y_t) \leq p_j, t = 1, \cdots, T\}|}{T}, \tag{B.6}$$

where $F_t(y_t)$ is the cumulative distribution function (CDF) and T is the number of samples in the dataset. A perfect calibrated regressor is expected to have $p_j = \hat{p}_j$ for all $j \in (1, \ldots, M)$.

However, the calibration curve only serves as visualization. A scalar representing the difference between p_j and $\hat{p}_j$ is more convenient for quantitative comparison. We therefore use the calibration error (CE, [3]) to quantify the miscalibration. It is defined as

$$\text{cal}(F_1, y_1, \cdots, F_T, y_T) \coloneqq \sum\nolimits_{j=1}^{M} \alpha_j \cdot (p_j - \hat{p}_j)^2, \tag{B.7}$$

where the scalars α_j are weights and $\alpha_j \equiv 1$ in our experiments.
Notations: We use $\doteq$ to indicate equality up to a constant. $p(x)$ is a probability density (or sometimes mass) function (pdf or pmf), and $p^0(x)$ is an unnormalized pdf. Hence, $\log p(x) \doteq \log p^0(x)$.

3 Proposed Approach

Our method is inspired by joint energy based (JEM) [10] for classification task, which reinterprets the logits entering the softmax layer as joint log-

likelihoods over inputs and target labels. Consequently, we propose to use a similar strategy to tackle regression-type problems. Thus, our goal is to learn the joint probability $p_\theta(x, y)$ (with trainable parameters θ) for regression tasks, where x and y are both continuous random variables. Unlike [12], only the conditional energy $E(y|x)$ is learned. Two architectures utilized in this work to represent $p_\theta(x, y)$ are described below. Due to the instabilities of the training process reported in [10], p_θ will be trained using a combination of a standard (maximum-likelihood) regression loss and an appropriate (sliced) score matching objective.

3.1 Energy-based regression models

JEM-Gaussian Let us start from a common regression architecture, where a neural network predicts a probability distribution, which is Gaussian in this case. A mean vector, denoted by $\phi(x)$, and a precision matrix, denoted by $\Lambda(x)$, are the network's output. In order to model $p(x)$ and entangle it with $p(y|x)$, the unnormalized joint model $p^0(x, y)$ is defined as

$$\log p^0(x, y) = -\tfrac{1}{2}(y - \phi(x))^\top \Lambda(x)(y - \phi(x)) + h(x). \tag{B.8}$$

where $h(x)$ is used to model the unnormalized probability of $\log p^0(x)$. Without the extra term $h(x)$, the marginal $p(x) = \int p(x, y)\, dy$ is constant. This is different to the classification setting, where the logits induce an unnormalized categorical distribution (and is independent of an additive bias), and consequently $\log p(x)$ is encoded in the respective bias. In the regression setting parametric and normalized continuous distributions are typically employed, which makes an explicit term $h(x)$ in the EBM above necessary. Marginalizing over y in $p^0(x, y)$ yields

$$\log p(x) \doteq h(x) - \tfrac{1}{2} \log\det \Lambda(x), \tag{B.9}$$

since integrating over a Gaussian random variable yields the partition function independent of x. In this model the conditional probability $p(y|x)$ is just a multivariate Gaussian distribution and therefore reads as

$$\log p(y|x) = -\tfrac{1}{2}(y - \phi(x))^\top \Lambda(x)(y - \phi(x)) - \log Z, \tag{B.10}$$

where

$$\log Z = \tfrac{D}{2} \log(2\pi) - \tfrac{1}{2} \log\det \Lambda(x). \tag{B.11}$$

D is the dimensionality of the output vector y. Without $h(x)$ in Eq. B.9, $\log p(x) = -\frac{1}{2}\log\det\Lambda(x)$ and an input sample x is considered more likely (according to the model) if its predictive uncertainty is *large*. This strongly couples the evidence $p(x)$ and the posterior $p(y|x)$ in a non-intuitive manner, and therefore the inclusion of a trainable mapping $h(x)$ is absolutely necessary.

JEM-Mixture Density Networks (JEM-MDN) For simplicity, we fist clarify the notation for MDN. The distribution of the MDN is assumed to be Gaussian. The weight, mean and standard deviation are denoted by $w_k(x)$, $\phi_k(x)$ and $\Sigma_k(x)$ for Gaussian component k. $\Lambda_k(x)$ is the precision matrix for Gaussian component k. The predictive distribution is modelled as follows,

$$p(y|x,k) = \mathcal{N}(y;\phi_k(x),\Sigma_k(x)) \qquad p(k|x) = w_k(x). \tag{B.12}$$

Because w_k represents the weight of each Gaussian component, $\sum_k w_k(x) = 1$ and it is achieved via

$$w_k(x) = \frac{\exp(f_k(x))}{\sum_{k'}\exp(f_{k'}(x))}, \tag{B.13}$$

where $f_k(x)$ represents the corresponding logit for the component k. Therefore,

$$\log w_k(x) = f_k(x) - \text{Softmax}_{k'}\, f_{k'}(x). \tag{B.14}$$

Hence

$$p(y,k|x) = p(y|k,x)p(k|x) = w_k(x)\mathcal{N}(y;\phi_k(x),\Sigma_k(x)). \tag{B.15}$$

Marginalizing over k consequently yields a mixture of Gaussians,

$$\begin{aligned} p(y|x) &= \sum_k w_k(x)\mathcal{N}(y;\phi(x),\Sigma_k(x)) \\ \log p(y|x) &= \text{Softmax}_k\Big(-\tfrac{1}{2}(y-\phi(x))^\top\Lambda_k(x)(y-g(x)) \\ &\quad -\log Z_k + \log w_k(x)\Big) \\ &= \text{Softmax}_k\Big(-\tfrac{1}{2}(y-\phi(x))^\top\Lambda_k(x)(y-\phi(x)) \\ &\quad -\log Z_k + f_k(x)\Big) - \text{Softmax}_{k'}\, f_{k'}(x). \end{aligned} \tag{B.16}$$

In order to entangle the $p(x)$ and each Gaussian component k, we use the unnormalized joint model

$$\log p(y, k, x) \doteq -\tfrac{1}{2}(y - \phi(x))^\top \Lambda_k(x)(y - \phi_k(x)) \tag{B.17}$$
$$- \log Z_k(x) + f_k(x) + h(x), \tag{B.18}$$

where Z_k is the respective normalization constant for each Gaussian, $h(x)$ is used to model the unnormalized probability of $\log p(x)$. According to $p(x) = \frac{p(y,k,x)}{p(y,k|x)}$,

$$\log p(x) = \log p(y, k, x) - \log p(y, k|x). \tag{B.19}$$

$\log p(y, k|x)$ can be obtained from Eq. B.15,

$$\log p(y, k|x) = \log w_k(x) - \log Z_k(x)$$
$$- \tfrac{1}{2}(y - \phi_k(x))^\top \Lambda_k(x)(y - \phi_k(x)).$$

By combining everything we arrive at the resulting log-marginal,

$$\log p(x) \doteq p^0(x) = h(x) + \text{Softmax}_k\, f_k(x). \tag{B.20}$$

Since both the logits $f_k(x)$ and $h(x)$ contribute to the marginal $p(x)$, in this setting it is possible to assume $h(x) = 0$. Recall that due to Eq. B.14, adding the same value to all logits $f_k(x)$ does not affect the mixture weights w_k, but will have an impact on $p(x)$. While in principle it is possible to set $h(x) \equiv 0$ w.l.o.g., we observed (and therefore report) slightly better results in our experiments when using a dedicated branch $h(x) \neq 0$.

3.2 Training loss

The training loss for our JEM on regression tasks reads as

$$\tfrac{1}{N} \sum\nolimits_i \log p_\theta(y_i|x_i) + \lambda \tilde{J}_{SSM}(x_i), \tag{B.21}$$

where $\lambda > 0$ is a hyper-parameter. Since score matching only yields unnormalized probabilities, $\log p_\theta(x)$ is only estimated up to an unknown constant (i.e. we only obtain its unnormalized variant p^0_θ). In our experiments we chose $\lambda = 1$.

The objective in Eq. (B.21) consists of two parts: first, it contains the negative log-likelihood loss, and the second term is the surrogate for the log-evidence. Under certain assumptions (in particular that the training data is sampled from $p_{\theta^*}(x, y)$, where θ^* are the true distribution parameters), the theory of proper scoring rules [23]–[25] asserts consistency of the maximizer $\hat{\theta} \to \theta^*$ for $N \to \infty$.

4 An Illustrative Example

We first demonstrate the effectiveness of our proposed method on the MNIST dataset of handwritten digits, which consists of 60k training images and 10k test images [26]. 10k of training images are used for validation. We recast the digit classification task as a regression problem and use LeNet5 [27] as the backbone to extract features in the following models:

Direct The extracted features $f_x \in \mathbb{R}^{84}$ are processed by two fully-connected layers ($84 \to 84$, $84 \to 1$) to output the prediction $\hat{y} \in \mathbb{R}$.

Gaussian f_x is processed by two heads of fully-connected layers ($84 \to 84$, $84 \to 1$) to obtain $\mu_\theta(x)$ and $\log \sigma_\theta^2(x)$.

JEM-Gaussian Feature extraction is followed by three heads of two fully-connected layers ($84 \to 84$, $84 \to 1$) to output $\mu_\theta(x)$, $\log \sigma_\theta^2(x)$ and $h(x)$.

Softmax $f_x \in \mathbb{R}^{84}$ enters two fully-connected layers ($84 \to 84$, $84 \to 84$, $84 \to$ C) to yield the logits for classes $\{0, 1, \cdots, 9\}$. It is trained via minimizing the the cross-entropy loss and L^2 loss, $J = J_{CE} + 0.1 J_{L^2}$.

JEM-Softmax The feature $f_x \in \mathbb{R}^{84}$ is processed by two heads of two fully-connected layers ($84 \to 84$, $84 \to$ C; $84 \to 84$, $84 \to 1$) to output the logits C for classes $\{0, 1, \cdots, 9\}$ and $h(x)$. It is trained via minimizing the CE loss, L^2 loss and SSM loss, $J = J_{CE} + 0.1 J_{L^2} + J_{SSM}$.

It can be seen in Table 1 that our method improves over two baseline methods in terms of the averaged MAE. Besides, it is shown in Fig. 1 that our model is consistently better calibrated compared with the Gaussian model.

Method	Direct	Gaussian	JEM-Gaussian (Ours)	Softmax	JEM-softmax (Ours)
MAE	0.1503 ± 0.0055	0.1598 ± 0.0036	0.1540 ± 0.0059	0.0371 ± 0.0025	**0.0364 ± 0.0009**

Table 1: Mean average error (MAE) on the MNIST test set.

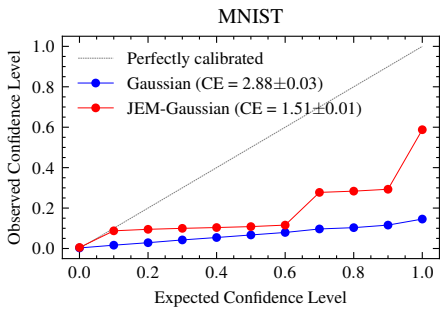

Figure 1: The obtained calibration curves and averaged calibration errors (CE) for the MNIST dataset.

5 Experiments

We apply our methods on three challenging computer vision regression tasks: object detection, age estimation and head pose estimation. Our proposed method is compared with both several standard regression methods and the state-of-art methods. All experiments are implemented in PyTorch. For age estimation and head post estimation tasks, we run 10 trials with 10 different random seeds.

5.1 Training

For age estimation and head pose estimation tasks, ResNet-50 [28] is used as the backbone. Both models are trained for 75 epochs using ADAM optimizer with learning rate 1e-4 and weight decay 0.0001. For object detection task, deep layer aggregation (DLA34) [29] is employed. The model is trained for 140 epochs (1X) with batch size 3 and learning rate 1.25e-4.

5.2 Evaluation Metrics

For age estimation and head pose estimation tasks, we use mean absolute error (MAE) and calibration error (CE) to evaluate the performance. For the Gaussian approach, it is straightforward to use the predicted mean as the prediction. For softmax baseline, the prediction is obtained by computing the softmax expected value. For the object detection experiments, we use the most popular metric-Average Precision(AP), in measuring the accuracy of object detectors. We follow the same protocol described in MSCOCO [30] and it is worth noting that AP is the average AP for IoU from 0.5 to 0.95 with a step size of 0.05, while AP50 and AP75 correspond to 0.5 and 0.75 IOU, respectively.

5.3 Age Estimation

UTKFace [31] dataset is used for age estimation task. It consists of human images labelled with ground truth ages. We employed the same dataset split as [12] and [32]. The subset of 16434 images are used, where only the ages are between 21 and 60 are selected. It is split with 3287 test images and 11503 images for training. Furthermore, the input images are cropped such that only faces are visible. The input image size is 200×200, which is the same as [12] and [32] for fair comparison.

Gaussian The feature $f_x \in \mathbb{R}^{2048}$ is firstly extracted from ResNet50 for the input images. The features vector f_x then is processed by two heads of two fully-connected layers ($2048 \to 2048$, $2048 \to 1$) to output the predicted mean $\mu(x)$ and $\log \sigma^2(x)$. The model is trained by minimizing the negative log-likelihood

$$J(\theta) = \frac{1}{n} \sum_{i=1}^{n} \frac{(y_i - \mu_\theta(x_i))^2}{\sigma_\theta^2(x_i)} + \log \sigma_\theta^2(x_i). \tag{B.22}$$

cEBM-Gaussian [12] The input images are firstly processed by ResNet50 to obtain features $f_x \in \mathbb{R}^{2048}$. Labels y is processed by four fully-connected layers to generate $g_y \in \mathbb{R}^{128}$. The two feature vectors g_x, g_y are concatenated together to for the feature $g_{x,y} \in \mathbb{R}^{2048+128}$, which is processed by two fully-connected layers ($2048 \to 2048$, $2048 \to 1$) to output $f_\theta(x, y) \in \mathbb{R}$. After training, the gradient ascent maximization of $f_\theta(x, y)$ is applied to refine the

prediction by the baseline models. Note our model does not have to be refined during inference stage compared with [12].

JEM-Gaussian The nput images are firstly processed by ResNet50 to obtain features $f_x \in \mathbb{R}^{2048}$ and and $f_h \in \mathbb{R}^{2048}$. After ResNet50, f_x is processed by two heads of two fully-connected layers (2048 $\rightarrow$ 2048, 2048 $\rightarrow$ 1) to output predicted mean $g(x)$ and $\log \Sigma(x)$. f_h is processed by one head of two fully-connected layers (2048 $\rightarrow$ 2048, 2048 $\rightarrow$ 1) to $h(x)$. The model is trained by minimizing the negative log joint probability. More specifically, the loss consists of two components. One is negative conditional log-likelihood loss which is defined in Eq. (B.22). The other is sliced score matching (SSM) loss used for $\log p(x)$.

Softmax We discretize the age to 40 classes $\{0, 1, \cdots, 39\}$. Input images are firstly processed by ResNet50 to obtain features $f_x \in \mathbb{R}^{2048}$. After ResNet50, f_x is processed by one head of two fully-connected layers (2048 $\rightarrow$ 2048, 2048 $\rightarrow$ C) to output the logits for 40 classes. The model is trained via minimzing the cross entropy loss and L^2 loss, $J = J_{CE} + 0.1 J_{L^2}$.

JEM-Softmax The input images are firstly processed by two ResNet50 to obtain features $f_x \in \mathbb{R}^{2048}$ and $f_h \in \mathbb{R}^{2048}$. After ResNet50, f_x is processed by one head of two fully-connected layers (2048 $\rightarrow$ 2048, 2048 $\rightarrow$ C) to output the logits for classes $\{0, 1, 2, \cdots, 39\}$. f_h is processed by one head of two fully-connected layers (2048 $\rightarrow$ 2048, 2048 $\rightarrow$ 1) to output $h(x)$. The model is trained by minimizing the negative log joint probability. More specifically, the loss consists of three components including the cross entropy loss, L^2 loss and sliced score matching (SSM) loss, $J = J_{CE} + 0.1 J_{L^2} + J_{SSM}$.

The results including the state-of-art methods are shown in Table **??**. Clearly, our method achieves much lower average MAE over the two baseline models, Gaussian and Softmax. Moreover, our model reduces the averaged MAE compared to the state-of-art [32]. As Fig. 2 shows, we also achieve lower CE compared to the Gaussian baseline. [33] achieves the lowest MAE (4.55 ± 0.04), however, it is not fair to compare because its training image size is 224×224, which is larger than [12], [32], and ours.

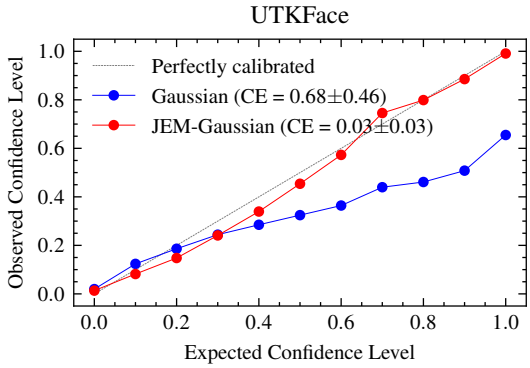

Figure 2: Calibration curves and the averaged calibration errors (CE) obtained for the UTKFace dataset.

models	[32]	Gaussian	cEBM-Gaussian [12]	JEM-Gaussian (Ours)	Softmax	JEM-Softmax (Ours)
MAE	5.47 ± 0.01	4.77 ± 0.04	4.66 ± 0.04	4.61 ± 0.03	4.78 ± 0.05	**4.61 ±0.01**

Table 2: Mean average error (MAE) in years for the age estimation task on the UTKFace [31] test dataset.

5.4 Head Pose Estimation

The BIWI dataset [34] is employed for head pose estimation task. It includes over 15K images of 20 people with their head pose, which is represented via the pitch, yaw and roll angles. Each angle is approximately distributed in the range $[-75°, 75°]$. We use the train-test split as the protocol 2 defined in [35], with 10163 images for training and 5065 images for testing. In addition, 1644 of the training images are used for validation. We utilize the same architecture as the age estimation task. The difference is that the dimensionality the prediction vector is increased to 3 because we need to predict the angle for pitch, yaw and roll, respectively. Besides, for Softmax and JEM-Softmax architectures, we discretize each angle to 151 classes $\{-75, -74, \cdots, 74, 75\}$ instead of 40 classes.

On average, it can be seen from Table **??** that our method again improves over the Gaussian baseline models in terms of both MAE and CE. In addition, we get slightly lower MAE compared with [12] in terms of the averaged MAE.

models	[36]	Gaussian	cEBM-Gaussian [12]	JEM-Gaussian (Ours)	Softmax	JEM-Softmax (Ours)
MAE	3.60 ± 0.08	3.12 ± 0.08	3.11 ± 0.07	**3.08 ± 0.05**	3.14 ± 0.05	3.29 ± 0.07

Table 3: Mean average error (MAE) in degrees for head pose estimation task on the BIWI [34] test dataset

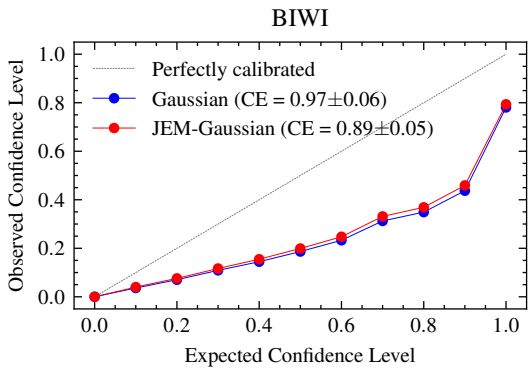

Figure 3: Calibration curves and the the averaged calibration errors (CE) for the BIWI dataset.

5.5 Object Detection

In the object detection experiment, we train CenterNet [37] and its variations including MDN [38] and ours on the Pascal VOC 2007 [39] dataset. There are 2501 images in the training set and 2510 images in the validation set. We follow the same protocol as MDN which only focuses on the object scale for a fairer comparison. As shown in Table **??**, it is quite obvious that involving JEM in training object detector is quantitatively beneficial. Further, using 3 components is generally better than using only 1 because the distribution of components can therefore closely follow that of the ground truth.

Method	Direct	Gaussian (Mixt.1)	JEM-Gaussian (Mixt.1)	MDN (Mixt.3)	JEM-MDN(Mixt.3)
AP	0.248	0.249	0.258	0.263	**0.284**
AP_{50}	0.480	0.491	0.497	0.506	**0.520**
AP_{75}	0.230	0.231	0.243	0.258	**0.285**

Table 4: Results for the object detection task on the Pascal VOC07 val set

6 Conclusion and Future Work

In this work we proposed joint energy-based models (EBMs) for regression tasks and how to train these models. The main motivation for using EBMs in this context is to establish better calibrated regression networks. In our experiments on challenging computer vision tasks, we demonstrate that our JEM-Gaussian and JEM-MDN approaches usually outperform a variety of regression baseline methods. In addition, our methods provide more accurate prediction uncertainties compared to baseline models. Hence, our conclusion is that using EBMs combined with appropriate training methods is beneficial to improve the performance for regression tasks.

We hypothesize that our models are also able to detect out-of-distribution samples, and a respective evaluation is subject of future work.

References

[1] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" In *Advances in Neural Information Processing Systems*, 2017.

[2] J. Gawlikowski, C. R. N. Tassi, M. Ali, *et al.*, "A survey of uncertainty in deep neural networks," *Artificial Intelligence Review*, pp. 1513–1589, 2023.

[3] V. Kuleshov, N. Fenner, and S. Ermon, "Accurate uncertainties for deep learning using calibrated regression," in *International Conference on Machine Learning (ICML)*, 2018, pp. 2801–2809.

[4] Y. LeCun, S. Chopra, R. Hadsell, F. J. Huang, and et al., "A tutorial on energy-based learning," MIT Press, 2006.

[5] G. Ruiqi, Y. Lu, J. Zhou, S. Zhu, and Y. Wu, "Learning generative convnets via multi-grid modeling and sampling," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[6] Y. Du and I. Mordatch, "Implicit generation and modeling with energy based models," in *Advances in Neural Information Processing Systems*, 2019.

[7] E. Nijkamp, M. Hill, T. Han, S.-C. Zhu, and Y. N. Wu, "On the anatomy of mcmc-based maximum likelihood learning of energy-based models," in *AAAI*, 2020.

[8] E. Nijkamp, M. Hill, S.-C. Zhu, and Y. N. Wu, "Learning non-convergent non-persistent short-run mcmc toward energy-based model," in *Advances in Neural Information Processing Systems*, 2019.

[9] G. Ruiqi, E. Nijkamp, D. Kingma, Z. Xu, A. Dai, and Y. Wu, "Flow contrastive estimation of energy-based models," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[10] W. Grathwohl, K.-C. Wang, J.-H. Jacobsen, D. Duvenaud, M. Norouzi, and K. Swersky, "Your classifier is secretly an energy based model and you should treat it like one," in *International Conference on Learning Representations (ICLR)*, 2020.

[11] M. Danelljan, L. Gool, and R. Timofte, "Probabilistic regression for visual tracking," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[12] F. K. Gustafsson, M. Danelljan, G. Bhat, and T. B. Schön, "Energy-based models for deep probabilistic regression," in *European Conference on Computer Vision (ECCV)*, 2020.

[13] F. Gustafsson, M. Danelljan, R. Timofte, and T. B. Schön, "How to train your energy-based model for regression," in *British Machine Vision Conference (BMVC)*, 2020.

[14] C. M. Bishop, "Mixture density networks," Tech. Rep., 1994.

[15] G. Mclachlan and K. Basford, *Mixture Models: Inference and Applications to Clustering*. 1988, vol. 38.

[16] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang, "A tutorial on energy-based learning," *Predicting structured data*, vol. 1, no. 0, 2006.

[17] A. Hyvärinen, "Estimation of non-normalized statistical models by score matching," *Journal of Machine Learning Research*, vol. 6, no. 4, pp. 695–709, 2005.

[18] Y. Song, S. Garg, J. Shi, and S. Ermon, "Sliced score matching: A scalable approach to density and score estimation," *Uncertainty in Artificial Intelligence*, pp. 574–584, 2020.

[19] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural computation*, vol. 23, no. 7, pp. 1661–1674, 2011.

[20] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton, "Regularizing neural networks by penalizing confident output distributions," in *International Conference on Learning Representations (ICLR)*, 2017.

[21] D. Levi, L. Gispan, N. Giladi, and E. Fetaya, "Evaluating and calibrating uncertainty prediction in regression tasks," in *Sensors*, 2020.

[22] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems*, 2017.

[23] T. Gneiting and A. E. Raftery, "Strictly proper scoring rules, prediction, and estimation," *Journal of the American statistical Association*, 2007.

[24] M. Parry, A. P. Dawid, and S. Lauritzen, "Proper local scoring rules," *The Annals of Statistics*, 2012.

[25] A. P. Dawid and M. Musio, "Theory and applications of proper scoring rules," *Metron*, 2014.

[26] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, 2012.

[27] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[28] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *British Machine Vision Conference (BMVC)*, 2016.

[29] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[30] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision (ECCV)*, Springer, 2014.

[31] Z. Zhang, Y. Song, and H. Qi, "Age progression/regression by conditional adversarial autoencoder," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[32] W. Cao, V. Mirjalili, and S. Raschka, "Rank consistent ordinal regression for neural networks with application to age estimation," *Pattern Recognition Letters*, 2020.

[33] A. Berg, M. Oskarsson, and M. O'Connor, "Deep ordinal regression with label diversity," in *International Conference on Pattern Recognition (ICPR)*, 2021.

[34] G. Fanelli, T. Weise, J. Gall, and L. Van Gool, "Real time head pose estimation from consumer depth cameras," in *Pattern Recognition*, 2011.

[35] T.-Y. Yang, Y.-T. Chen, Y.-Y. Lin, and Y.-Y. Chuang, "Fsa-net: Learning fine-grained structure aggregation for head pose estimation from a single image," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[36] Z. Mi, Y. Luo, and W. Tao, "Ssrnet: Scalable 3d surface reconstruction network," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[37] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[38] A. Varamesh and T. Tuytelaars, "Mixture dense regression for object detection and human pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[39] M. Everingham, "The pascal visual object classes challenge 2007," 2009.

PAPER C

GEN: Pushing the limits of softmax-based out-of-distribution detection

Xixi Liu, Yaroslava Lochman, Christopher Zach

IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)
pp. 23946-23955, 2023
©DOI: 10.1109/CVPR52729.2023.02293

The layout has been revised.

Abstract

Out-of-distribution (OOD) detection has been extensively studied in order to successfully deploy neural networks, in particular, for safety-critical applications. Moreover, performing OOD detection on large-scale datasets is closer to reality, but is also more challenging. Several approaches need to either access the training data for score design or expose models to outliers during training. Some post-hoc methods are able to avoid the aforementioned constraints, but are less competitive. In this work, we propose Generalized ENtropy score (GEN), a simple but effective entropy-based score function, which can be applied to any pre-trained softmax-based classifier. Its performance is demonstrated on the large-scale ImageNet-1k OOD detection benchmark. It consistently improves the average AUROC across six commonly-used CNN-based and visual transformer classifiers over a number of state-of-the-art post-hoc methods. The average AUROC improvement is at least 3.5%. Furthermore, we used GEN on top of feature-based enhancing methods as well as methods using training statistics to further improve the OOD detection performance. The code is available at: `https://github.com/XixiLiu95/GEN`.

1 Introduction

In order to make the usage of deep learning methods in real-word applications safer, it is crucial to distinguish whether an input at test time is a valid in-distribution (ID) sample or a previously unseen out-of-distribution (OOD) sample. Thus, a trained deep neural network (DNN) should ideally know what it does not know [3]. This ability is particularly important for high-stake applications in autonomous driving[4] and medical image analysis[5]. However, it is common for neural networks to make overconfident predictions even for OOD samples. A recent survey on OOD detection [6] identifies several scenarios requiring the detection of OOD samples, with covariate shift (change in the input distribution) and semantic shift (change in the label distribution) being two important settings.

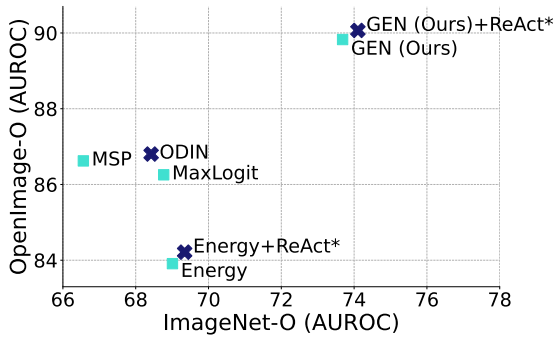

Figure 1: *Performance of Post-hoc OOD Detection Methods Applied to 6 Classifiers Trained on ImageNet-1K.* Reported are AUROC values (%) averaged over the models. Methods marked with light squares use information from logits / probabilities. Methods marked with dark crosses also use information from features. ReAct* corresponds to performing extra feature clipping before computing the score.

In this work, we focus on the semantic shift scenario, meaning that we aim to detect inputs with semantic labels not present in the training set. When solving the OOD detection problem, the idea is to design a scalar score function of a data sample as an argument that assigns higher values to true ID samples. The semantic shift scenario also allows us to mainly focus on the predictive distribution as provided by a DNN classifier to design such score function.

A number of existing works for OOD detection rely on the predictive distribution [1], [7], but often a better OOD detection performance can be achieved when also incorporating feature statistics for ID data [8]–[12]. These high-performing methods have practical constraints that can be challenging to eliminate: some methods require access to at least a portion of training data [8], [9], [11], [12] while others need access to internal feature activations [10]. However, commercially deployed DNNs are often black-box classifiers, and the training data is likely to be confidential. Hence, the goal of this work is to explore and push the limits of OOD detection when the output of a softmax layer is the only available source of information. Our method therefore falls under the *post-hoc* category of OOD detection frameworks, where only a trained DNN

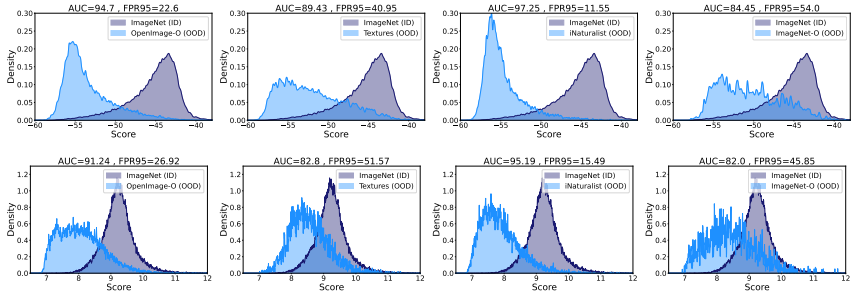

Figure 2: *Score Distributions.* The top row is GEN, and the bottom one is Energy [1]. The distributions are shown for the ID ImageNet-1K dataset (dark blue) and four OOD datasets (light blue). The classification model used here is Swin [2].

is used without the need for training data. Fig. 1 highlights its performance compared to other methods in this category.

Contribution We propose GEN, a simple but effective entropy-based method for OOD detection. (i) GEN uses predictive distribution only. It does not require re-training and/or outlier exposure, it does not use any training data statistics. (ii) Yet it performs very well (see Figs. 1 and 2), meaning that it can potentially be used in more constrained model deployment scenarios. Compared to other post-hoc methods, score distributions produced by GEN lead to a better ID/OOD separation. We show that our method consistently achieves better results in terms of AUROC (and usually in terms of FPR95) compared to other post-hoc methods. In particular, GEN on average outperforms other post-hoc methods on the largest and carefully constructed OOD dataset OpenImage-O as well as on the very challenging ImageNet-O dataset based on natural adversarial examples.

2 Related Work

Score design Given a pre-trained softmax neural classifier, designing a proper score function that aims to separate ID from OOD data is essential to successfully perform OOD detection. [7] proposes the maximum predicted softmax

Method	Equation	Free of		Space		
		ID train data	ID labels	features	logits	probs
MSP [7]	$\max_c p_c$	✓	✓			✓
MaxLogit [9] / Energy [1]	$\max_c f_c(\mathbf{z})$ / $\text{LogSumExp}\, f(\mathbf{z})$	✓	✓		✓	
GradNorm [13]	$\|\mathbf{p} - 1/C\|_1 \cdot \|\mathbf{z}\|_1$	✓	✓	✓		✓
ODIN [14]	$\tilde{\mathbf{x}} = \mathbf{x} + \varepsilon \text{sign}\left(\nabla_{\mathbf{x}} \log \max_c p_c(\mathbf{x})\right)$	✓	✓			✓
ReAct [10]	$\tilde{\mathbf{z}} = \min(\mathbf{z}, b)$	✗, b	✓	✓		
RankFeat [12]	$\tilde{\mathbf{o}} = \mathbf{o} - s_1 \mathbf{u}_1 \mathbf{v}_1^\top$	✓	✓	✓		
Mahalanobis [8]	$\max_c -(\mathbf{z} - \hat{\mu}_c)^\top \hat{\Sigma}^{-1} (\mathbf{z} - \hat{\mu}_c)$	✗, $\hat{\Sigma}, \hat{\mu}_c$	✗	✓		
pNML [15]	$\log \sum_{c=1}^{C} \frac{p_c}{p_c + p_c^\kappa (1 - p_c)}, \quad \kappa = \frac{\mathbf{z}^\top \Sigma_{\text{corr}} \mathbf{z}}{1 + \mathbf{z}^\top \Sigma_{\text{corr}} \mathbf{z}}$	✗, Σ_{corr}	✓			
KL Matching [9]	$-\min_c D_{\text{KL}}(\mathbf{p} \parallel \mathbf{d}_c)$	✗, $\mathbf{d}_c$	✗			✓
Residual [11]	$-\|\mathbf{z}^{P^\perp}\|_2$	✗, P	✓	✓		
ViM [11]	$-\alpha \|\mathbf{z}^{P^\perp}\|_2 + \text{LogSumExp}\, f(\mathbf{z})$	✗, α, P	✓	✓	✓	
Shannon Entropy	$-\sum_{m=1}^{M} p_{i_m} \log p_{i_m}, \quad p_{i_1} \geq \cdots \geq p_{i_C}, \ \gamma \in (0, 1)$	✓	✓			✓
GEN	$G_\gamma(\mathbf{p}) = -\sum_{m=1}^{M} p_{i_m}^\gamma (1 - p_{i_m})^\gamma, \quad p_{i_1} \geq \cdots \geq p_{i_C}, \ \gamma \in (0, 1)$	✓	✓			✓
GEN + ReAct [10]	$G_\gamma\left(\text{Softmax}(f(\tilde{\mathbf{z}}))\right), \quad \tilde{\mathbf{z}} = \min(\mathbf{z}, b)$	✗, b	✓	✓		✓
GEN + Residual [11]	$G_\gamma(\mathbf{p}) \cdot \|\mathbf{z}^{P^\perp}\|_2$	✗, P	✓	✓		✓

Table 1: *Technical Comparison of OOD Detection Methods.* $\mathbf{x}$ is an input, $\mathbf{z}$ is an output of the penultimate layer (also called features), $f(\mathbf{z})$ denotes logits, $\mathbf{p} = \text{Softmax}(f(\mathbf{z}))$ is predictive distribution, and C is the number of classes. Enhancing methods work in the input / feature space, i.e., they perturb original inputs $\mathbf{x}$, features $\mathbf{z}$, or intermediate convolutional features $\mathbf{o}$ (where the perturbed result of e.g. $\mathbf{x}$ is $\tilde{\mathbf{x}}$). Several methods require pre-computation of training data statistics. In particular, Mahalanobis [8] needs the empirical per-class mean $\hat{\mu}_c$ and tied covariance $\hat{\Sigma}$ of the training features. pNML [15] needs the empirical correlation matrix Σ_{corr}. KL Matching [9] requires the knowledge of per-class predictive distributions $\mathbf{d}_c$. Residual and ViM [11] require the principal space P of the training features. Our method GEN uses information from the probability space only, does not perturb the inputs nor does it need ID data.

probability (MSP) and thereby establishes an initial baseline for such scores. Subsequently, [8] defines the score as the minimum Mahalanobis distance between features and the empirical class-wise centroids, which are computed from training samples. The energy score is suggested in [1] and is computed via LogSumExp, which is the soft maximum of the logits. This energy score can also be understood as the unnormalized log data density [16]. Unlike [1], [9] proposes the (hard) maximum of the logits as OOD score. [9] also gives a statistics-based alternative called KL Matching, where the posterior distribution template $\bar{p}_k$ for each class k is computed from training data. At test time, the KL divergence between the predictive distribution and each posterior distribution template is calculated for a given sample. The negative minimum KL divergence is taken as the score.

Previous methods use information from one of the spaces, feature, logit, or predictive distribution. GradNorm [13] incorporates the information from both feature and predictive distribution. Specifically, this score is a product of the feature norm and the distance from the predictive to the uniform distribution. Predictive Normalized Maximum Likelihood (pNML) [15] derives a score function based on the generalization error (the regret), which needs to access the empirical correlation matrix of training features and the predictive distribution. ViM [11] uses information from all spaces via introducing a virtual logit with corresponding rescaling factor α. First, the residual of the feature $\mathbf{z}$ is calculated as $\|\mathbf{z}^{P^\perp}\|$, where P is the so-called principal space (i.e. the principal component of the features). A mixing coefficient α is computed in order to match the scale of the virtual logits to the real maximum logits over the training set. The final score is calculated as the softmax probability of the virtual logit and can be also interpreted as a combination of the energy score LogSumExp $f(\mathbf{z})$ and rescaled residual $-\alpha\|\mathbf{z}^{P^\perp}\|$. Our GEN score can actually replace the energy in this formulation to further improve the OOD detection performance.

Score enhancing methods There is also a line of research that aims to enhance the OOD detection performance for given score functions [10], [12], [14], [17]. ODIN [14] uses a temperature scaling T for logits and adds perturbation to the input sample to enhance the reliability of OOD detection when MSP score is used. Specifically, each logit is divided by a temperature T, and the perturbed input can be calculated as $\tilde{\mathbf{x}} = \mathbf{x} + \varepsilon \text{sign}(\nabla_{\mathbf{x}} \log S_{\hat{y}}(\mathbf{x}; T))$, where $S_{\hat{y}}(\mathbf{x}; T)$ is the maximum softmax probability. However, T needs to be tuned with OOD samples. Generalized ODIN [17] aims to free ODIN [14] from the need of OOD samples without decreasing the OOD performance. ReAct [10] applies feature clipping on the penultimate layer of neural networks. Specifically, an operation $\min(f(\mathbf{z}), c)$ is applied element-wise to the feature vector $f(\mathbf{z})$. This enhancing method is compatible with MSP score [7] and energy score [1]. RankFeat [12] looked into the distribution of the singular values for ID and OOD samples and found that OOD samples appear to have larger dominant singular values than ID samples. Instead of using the largest singular value as the score, they remove the rank-1 matrix $s_1\mathbf{u}_1\mathbf{v}_1^\top$ composed of the largest singular value s_1 and the associated singular vectors $\mathbf{u}_1$, $\mathbf{v}_1$ from the intermediate (flattened) feature maps $\mathbf{o}$. The modified features $\tilde{\mathbf{o}}$ are pro-

cessed by the remaining part of the neural network, and the energy score is computed. A summary of the aforementioned score design and enhancing methods is given in Table 1.

Modifying the training loss An alternative to the OOD detection score design for fixed networks is to incorporate the OOD samples into the training procedure. Specifically, adding a separate network head (and a suitable loss) for confidence prediction [18], reinterpreting logits as joint log-probabilities (over inputs and labels) and training using a log-evidence term in addition to the standard cross-entropy loss [16], or incorporating a subspace prior on features [19] are approaches to obtain DNNs better suited for OOD detection (besides solving a classification task). [20] addresses the fine-grained classification setting in particular and leverages semantic groups (and a dedicated out-of-group label), which simplifies decision boundaries and therefore helps to identify OOD samples. It is further possible to explicitly include OOD data into the training phase of a DNN. Joint minimization of a classification loss (over ID data) and a regularization term favoring highly uncertain predictive distributions for OOD data is suggested in [21], [22].

Classifier calibration Supervised training usually leads to uncalibrated classifiers, which tend to be either over-confident (usually) or under-confident (rarely) in their prediction confidence. In short, "a predicted probability (vector) should match empirical (observed) accuracy" [23]. The calibration of a pre-trained classifier can be improved by post-processing the logits [24]–[26] or by using classifier ensembles [27]. Since a number of OOD detection approaches uses solely the logits or resulting predictive distribution as input, the OOD detection performance may vary between the trained vanilla and the calibrated classifiers. At least for monotone transformations of logits [24], [25] the performance of MSP [7], MaxLogit [9], and Energy [1] scores should be unaffected (in terms of AUROC). Other OOD detection scores (e.g. GradNorm) will be affected.

Notation The penultimate layer output is denoted as $\mathbf{z}$, which is the feature vector occurring immediately before the logit layer. The vector of logits is $f(\mathbf{z})$ and is typically computed via a linear layer, $f(\mathbf{z}) = \mathbf{W}\mathbf{z}+\mathbf{b}$ for a weight matrix $\mathbf{W}$ and bias vector $\mathbf{b}$. The output of a classifier network is the predictive

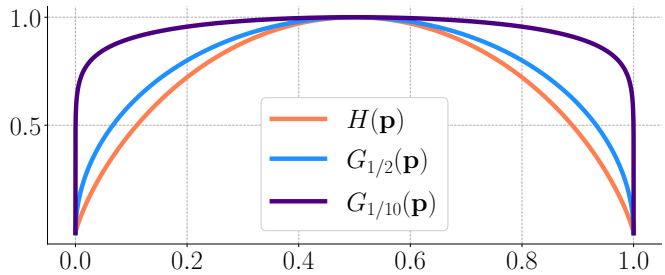

Figure 3: Generalized entropies: $G_{1/2}$, $G_{1/10}$ and the Shannon entropy $H(p)$ for a Bernoulli random variable (all scaled to the same range).

distribution $\mathbf{p} = \text{Softmax}(f(\mathbf{z}))$. Categorical distributions over C classes are elements of the C-dimensional unit simplex Δ^C. Equality up to an irrelevant constant is denoted by $\doteq$.

3 Generalized Entropy Score

Our aim is to rely solely on the logits and in particular on the predictive distribution as much as possible for OOD detection, because relatively simple scores using only this information are performing surprisingly well [1], [7], [9], [13]. Further, such an approach is agnostic to any information on the classifier training, the training set, or explicit OOD samples. The backbone of a classifier can be even a black box computation. Finally, the neural collapse hypothesis [28] states that the features from the penultimate layer have very limited additional information compared with the logits.

Our main assumption is, that the training loss for a classifier is dominated by a term that is minimal for a "pure" one-hot predictive distribution, which is a valid assumption for a wide range of losses (such as cross-entropy, squared Euclidean loss, label smoothing loss [29], focal loss [30] and more). Hence, ID test samples close to the training data are expected to result in a confident prediction. The prediction confidence can be measured in a variety of ways, and a statistical distance to either the uniform distribution or to a one-hot distribution. Common statistical distances are in the f-divergence family (e.g. [31]), Wasserstein metric [32], [33] and the total variation distance.

Here we borrow the concept of generalized entropy from the literature on

proper scoring rules [34], [35]: a *generalized entropy* G is a differentiable and concave function on the space of categorical distributions Δ^C. The Bregman divergence $D_G(\mathbf{p}\|\mathbf{q})$ between 2 elements $\mathbf{p}, \mathbf{q} \in \Delta^C$ is the linearization error

$$D_G(\mathbf{p}\|\mathbf{q}) := G(\mathbf{q}) - G(\mathbf{p}) + (\mathbf{p} - \mathbf{q})^\top \nabla G(\mathbf{q}), \tag{C.1}$$

which is non-negative for concave G. We assume that G is invariant under permutations of the elements in $\mathbf{p}$ (all class labels are treated equally). Now the Bregman divergence between $\mathbf{p}$ and the uniform categorical distribution $\mathbf{u} = \mathbf{1}/\mathbf{C}$ reduces to the negated generalized entropy (up to additive constants),

$$\begin{aligned} D_G(\mathbf{p}\|\mathbf{u}) &= G(\mathbf{u}) - G(\mathbf{p}) + (\mathbf{p} - \mathbf{u})^\top \nabla G(\mathbf{u}) \\ &\doteq -G(\mathbf{p}) + \underbrace{(\mathbf{p} - \mathbf{u})^\top \nabla G(\mathbf{u})}_{=0}. \end{aligned} \tag{C.2}$$

The last term vanishes since $\nabla G(\mathbf{u}) = \nabla G(\mathbf{1}/\mathbf{C}) = \kappa \mathbf{1}$ (for some $\kappa \in \mathbb{R}$, using our assumption of permutation invariance for G) and therefore $(\mathbf{p} - \mathbf{u})^\top \nabla G(\mathbf{u}) \propto \mathbb{E}_{\mathbf{p}}[\kappa] - \mathbb{E}_{\mathbf{u}}[\kappa] = 0$. Overall, using a negated entropy as score can be interpreted as a statistical distance between the predictive distribution $\mathbf{p}$ and the uniform distribution $\mathbf{u}$.

Our particular attention is on the following family of generalized entropies,

$$G_\gamma(\mathbf{p}) = \sum\nolimits_j p_j^\gamma (1 - p_j)^\gamma \tag{C.3}$$

for a $\gamma \in (0, 1)$. It is straightforward to verify that the mapping $p \mapsto p^\gamma (1-p)^\gamma$ is concave in the domain $[0, 1]$ for all $\gamma \in [0, 1]$. The choice $\gamma = 1/2$, i.e.

$$G_{1/2}(\mathbf{p}) = \sum\nolimits_j \sqrt{p_j (1 - p_j)}, \tag{C.4}$$

is connected to the (non-robust) exponential loss occurring in the boosting method (as detailed in [36]), and therefore considered to be more sensitive than e.g. the Shannon entropy $H(\mathbf{p}) = -\sum_j p_j \log p_j$[1]. Lower values of γ amplify this behavior: Fig. 3 depicts the graphs of H, $G_{1/2}$ and $G_{1/10}$ for a Bernoulli random variable with parameter p. In particular the entropy $G_{1/10}$ increases rapidly near $p = 0$ and $p = 1$. Hence, $G_{1/10}$ can be seen as very sensitive detector for uncertainties in the predictive distribution.

[1]The regular Shannon entropy in analogy leads to the soft-plus loss in logistic regression.

To sum up, the motivation behind GEN is simple and straightforward. The aim of using a generalized entropy is to amplify minor deviations of a predictive distribution from the ideal one-hot encoding. In practice, this high sensitivity turns out to require some degree of robustness (and numerical stability) in the fine-grained classification setting, which we achieve by "trimming" the predictive distribution described next.

Truncation If we consider sorted predictive probabilities, $p_{j_1} \geq p_{j_2} \geq \cdots \geq p_{i_C}$, then the generalized entropy G_γ as a sum over all classes can be dominated by the tail, i.e. the large fraction of very small probabilities. Random but small variations in those probabilities have a significant impact on the score. With growing C, extremely small but random tails can change the sort order of discrete probabilities w.r.t. the generalized entropy. Hence, the ability of generalized entropies to discriminate finely between probability vectors near the boundary (compared to the regular Shannon entropy) comes at a cost in the many-class setting. Using a truncated sum over the top-M classes made G_γ robust in synthetic setups. Overall, our score is designed to capture small entropy variations in the top-M classes.

4 Experiments

OOD detection benchmarks have matured over the years—there has been a transition from small scale datasets such as CIFAR-10, CIFAR-100 to more realistic large-scale dataset such as ImageNet-1K[37] and OpenImage-O[38], and the evaluation metrics have converged to AUROC and FPR95 values. We follow the recent development in evaluation strategy which we describe in Sec. 4. In our experiments, we closely follow the large-scale evaluation protocol conducted in ViM [11]. In particular, the choice of discriminative models with officially released pre-trained weights as well as the large-scale ID / OOD datasets. Note that all the methods studied in this work are deterministic.

Models We used several commonly-used convolutional and transformer-based architectures for large-scale image classification. These include Big Transfer [39], Vision Transformer [40], RepVGG[41], ResNet-50-D[42], DeiT[43], and Swin[2]. Big Transfer (BiT) [39] refers to the set of large neural network

architectures and techniques (such as large batches, group normalization and weight standardization) for an efficient transfer learning and improved generalization. We utilized a variant with ResNet-v2-101 (BiT-S-R101x1 checkpoint). Vision Transformer (*ViT*)[40] is a pure transformer-based model for image classification. Its input image is cut into a sequence of patches with corresponding position embeddings. We use ViT-B/16 version in our experiments. *RepVGG*[41] model combines VGG and ResNet architectures in a way that allows for structural re-parameterizations. In particular, RepVGG is turned from a multi-branch ResNet-like network topology (used for training) into a plain VGG-like architecture with only 3×3 convolutions (used for inference). *ResNet-50-D* is one of the refined versions of ResNet architecture proposed by [42] to improve its performance. Shifted WINdows (*Swin*) transformer[2] injects priors coming from vision, such as hierarchy, locality and translational invariance, into a vision transformer network. Data-efficient image Transformers (*DeiT*)[43] is a token-based strategy for transformer distillation that enables efficient training and produces competitive results on downstream tasks. Specifications of the aforementioned architectures are summarized in Table 2.

Classifier	Feat.	Top-1 (%)	Params
BiT-S-R101x1[39]	2048	81.30	44.54M
BiT-S-R101x1[39] (ckpt [13])	2048	75.19	44.54M
ViT-B/16 [40]	768	85.43	86.86M
RepVGG-B3[41]	2560	80.52	120.52M
ResNet-50-D[42]	2048	80.52	23.53M
DeiT-B/16[43]	768	81.98	85.80M
SWIN-B/4[2]	1024	85.27	86.74M

Table 2: Specifications of different architectures: dimensionality of the feature (penultimate layer output) space, top-1 accuracy on ImageNet-1K validation dataset, and the number of parameters.

Datasets We perform OOD detection on a large-scale OOD detection benchmark with ImageNet-1K[37] as ID dataset. We evaluate our methods using four commonly-used OOD datasets, which include OpenImage-O [11], Texture [44], iNaturalist [45], and ImageNet-O [46]. These datasets cover different domains including fine-grained images, scene images, textures images,

etc.. In particular, ImageNet-O consists of natural adversarial examples that are unforeseen classes in ImageNet-1K and cause model's performance to significantly degrade. OpenImage-O is the largest OOD dataset for ImageNet-1K released by ViM [11]. The authors discover that previous datasets like SUN [47], Places [48], and Texture [44] have a subset of images that is indistinguishable from ID data and thus manually select images from OpenImage-v3 dataset[38] that are OOD w.r.t. ImageNet-1K. Specifications of the used datasets are summarized in the supplementary material.

Post-hoc methods First and foremost, we compare GEN to the scores within the same family of post-hoc methods, *i.e.* not requiring prior access to the training dataset with or without labels. The first group of methods includes MSP [7], MaxLogit [9], Energy [1], and GradNorm [13] that operate on the output space. In addition, the score function that uses negative Shannon entropy is also considered. The second group comprises input / feature enhancing methods like ODIN [14] and ReAct*. ReAct* is a local version of ReAct [10] that clips penultimate activations of the current sample based on the values alone. We furthermore combine GEN with ReAct* to achieve better performance.

Methods requiring ID train data One of the advantages of GEN is that it does not require ID training dataset. Nevertheless, when the training data is available, it is potentially beneficial to combine this information with GEN (see Table. 1). We compare it to existing methods that require pre-computation of the training data statistics, such as KL Matching [9], Mahalanobis [8], pNML [15], Residual, and ViM [11].

Evaluation metrics We use two standard evaluation metrics for OOD detection. The first one is the area under the receiver operating characteristic curve (AUROC), for which higher values indicate better performance. The second one is FPR95 — the false positive rate when the true positive rate is 95%. Lower FPR95 values are better. The reported units for both metrics in all tables are percentages.

4.1 OOD Detection Performance Results

In this section, the results of the OOD detection benchmark are presented. We reproduce the results for all methods (except for ODIN [14]) and obtain slightly different results than reported in [11]. In our experiments, we used NVIDIA GeForce RTX 3080, CUDA 11.5 + PyTorch 1.11.

	Classifier + OOD Method	**OpenImage-O**		**Textures**		**iNaturalist**		**ImageNet-O**		**Average**	
		AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓
	BiT-S-R101x1										
Post-hoc	MSP [7]	83.05	76.21	79.76	77.13	87.90	64.53	57.16	96.90	76.97	78.69
	MaxLogit [9]	82.33	79.75	81.65	73.59	86.78	70.52	62.99	96.90	78.44	80.19
	Energy [1]	80.59	82.00	81.10	73.91	84.52	74.93	63.56	96.35	77.44	81.80
	GradNorm [13]	70.68	79.34	**83.12**	**55.72**	86.13	**58.34**	53.73	**91.90**	73.42	**71.33**
	ODIN [14]	**85.64**	**72.83**	81.60	74.07	86.73	70.75	63.00	96.85	79.24	78.63
	ReAct*	80.83	81.85	81.44	73.74	84.77	74.80	63.63	96.30	77.67	81.67
	Shannon Entropy	83.98	80.48	81.30	76.32	88.73	69.66	60.42	97.30	78.61	80.94
	GEN	83.77	80.43	81.48	77.93	88.67	68.32	66.09	97.30	80.00	81.00
	GEN + ReAct*	83.99	80.35	81.80	77.87	**88.90**	68.03	**66.18**	97.25	**80.22**	80.88
Require ID	KL Matching [9]	87.94	54.92	86.91	50.89	92.95	**33.19**	65.76	86.80	83.39	56.45
	Mahalanobis [8]	82.62	66.24	97.33	13.95	85.79	64.71	80.37	70.20	86.53	53.77
	ReAct [10]	85.43	67.45	90.65	50.14	91.50	48.65	67.04	91.50	83.66	64.44
	pNML [15]	88.62	55.27	93.59	22.25	**93.12**	38.21	67.27	86.35	85.65	50.52
	Residual [11]	80.20	68.05	97.67	11.14	76.93	80.18	81.58	65.60	84.09	56.24
	ViM [11]	89.96	49.01	**98.92**	**4.63**	89.38	55.09	83.85	**61.25**	90.53	42.50
	GEN + ReAct [10]	85.36	78.22	84.68	74.09	90.27	62.36	67.54	97.10	81.96	77.94
	GEN + Residual [11]	**91.75**	**43.83**	98.54	5.78	92.25	47.13	**83.88**	63.70	**91.61**	**40.11**
	Swin										
Post-hoc	MSP [7]	91.38	34.81	85.31	51.74	94.76	22.97	78.86	63.90	87.58	43.36
	MaxLogit [9]	92.09	26.70	84.81	47.23	95.71	15.34	81.07	52.10	88.42	35.34
	Energy [1]	91.24	26.92	82.80	51.57	95.19	15.49	82.00	**45.85**	87.81	34.96
	GradNorm [13]	45.52	77.94	37.12	93.02	33.79	88.81	50.27	78.05	41.68	84.45
	ODIN [14]	91.38	28.42	85.74	44.59	94.24	19.65	80.62	53.65	88.00	36.58
	ReAct*	91.23	26.98	82.79	51.69	95.18	15.50	82.00	45.90	87.80	35.02
	Shannon Entropy	93.16	25.61	87.15	43.84	95.95	16.21	82.13	51.95	89.60	34.40
	GEN	**94.70**	**22.60**	**89.43**	**40.95**	**97.25**	**11.55**	**84.45**	54.00	**91.46**	**32.28**
	GEN + ReAct*	94.69	22.62	89.42	41.01	**97.25**	11.56	84.44	54.00	91.45	32.30
Require ID	KL Matching [9]	91.86	39.93	86.82	53.24	94.75	27.76	81.78	67.30	88.80	47.06
	Mahalanobis [8]	94.35	34.85	89.95	49.09	98.69	5.38	85.43	73.65	92.11	40.74
	ReAct [10]	93.71	22.61	85.62	47.79	97.49	9.99	83.83	**44.95**	90.16	31.34
	pNML [15]	95.53	**19.29**	91.55	**33.29**	97.84	8.98	87.22	45.05	93.03	**26.65**
	Residual [11]	94.44	33.40	91.36	43.26	98.90	4.79	86.66	68.65	92.84	37.53
	ViM [11]	**95.93**	24.43	**92.40**	37.98	**99.29**	**2.62**	**88.74**	59.00	**94.09**	31.01
	GEN + ReAct [10]	94.80	22.23	89.47	40.85	97.42	10.67	84.48	54.25	91.54	32.00
	GEN + Residual [11]	95.73	25.06	92.23	37.66	99.13	3.10	88.07	61.50	93.79	31.83

Table 3: *Per-Dataset Performance of OOD Detection Methods.* The classifiers are BiT [39] and Swin [2]. The ID dataset is ImageNet-1K, the OOD datasets are OpenImage-O, Textures, iNaturalist and ImageNet-O. For GEN, the number of maximal logits is set to 10% and $\gamma = 0.1$. Clipping quantile for ReAct* is set to 0.9995, and for ReAct [10] — to 0.999. The best performing method is in bold, the second best is underlined.

Results on BiT and Swin We show detailed results on BiT [39] and Swin [2] architectures, since BiT is commonly used for large-scale OOD detection [11]–[13], [20] and Swin [2] is the recent transformer-based architecture.

The results of BiT [39] are presented in the top half of Table 3. First, one can see from the "Post hoc" block that our score achieves the highest average AUROC (across four datasets) compared to other post-hoc methods. In particular, we obtain the highest AUROC on ImageNet-O and iNaturalist. Furthermore, using feature clipping further improves the performance in terms of AUROC and FPR95. For this classifier, GradNorm [13] gives lower FPR95 values. We think this could be connected to the lower classification accuracy of the pre-trained models (see Tab. 2) and/or model specifics because GradNorm performs significantly worse for other classifiers (see Tab. 4 and Tab. 2 in Supplementary). Then we look into the methods using ID data statistics. Our score is combined with ReAct [10] and Residual [11] methods, which compute compressed information of feature space from all training data. Results from the "Require ID" block show that using information from feature space could further improve our score. Specifically, our method combined with Residual [11] achieves the state-of-the-art results on in terms of the averaged AUROC and FPR95 (over four datasets) when using BiT [39], in particular on the challenging OpenImage-O dataset.

The results of Swin [2] are shown in the bottom half of Table 3. Results from the "Post-hoc" block show that our score is consistently better in terms of AUROC values than all other post-hoc methods. Particularly, our method outperforms MaxLogit [9] by 3% on average in terms of AUROC. According to the "Require ID" block, our performance is comparable to ViM [11].

To visualize OOD performance, we present the score distributions using our score and Energy [1] score in Figure 2 and it shows that our method makes ID/OOD separation better. Interestingly, the score distribution drawn based on our score function is smoother.

Averaged results for other architectures To further investigate the effectiveness and robustness of our score, we perform OOD detection on four remaining architectures, RepVGG [41], ResNet-50-D [42], ViT [40], and DeiT [43]. The averaged results over four datasets are shown in Table 4. First, it is apparent that our score continually gains the best AUROC on different architectures compared to all post-hoc methods. Specifically, our method outperforms

MSP [7] on average with a notable margin, almost 5%. Moreover, our score also obtains the lowest averaged FPR95 over four datasets and four architectures. It is significantly better than all other post-hoc methods in terms of FPR95, with a nearly 4% margin. The results of combining our score with information from ID dataset can be found in the bottom half of the Table 4. It shows that our method achieves competitive results compared with ViM [11].

	OOD Method	RepVGG [41]		ResNet-50-D [42]		ViT [40]		DeiT [43]		Average	
		AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓
Post-hoc	MSP [7]	78.02	70.83	77.99	68.10	89.33	41.89	79.44	66.22	81.19	61.76
	MaxLogit [9]	77.47	73.55	75.47	69.28	94.56	24.34	76.77	64.37	81.07	57.89
	Energy [1]	76.29	79.11	71.25	78.01	**94.89**	22.54	72.81	69.88	78.81	62.39
	GradNorm [13]	52.98	94.98	44.04	96.08	90.32	28.66	32.05	97.47	54.85	79.30
	ODIN [14]	77.72	72.68	75.27	68.56	94.57	24.25	77.13	63.92	81.17	57.35
	ReAct*	77.60	78.57	71.55	77.70	**94.89**	**22.83**	72.82	69.87	79.21	62.24
	Shannon Entropy	79.01	71.81	78.82	66.41	91.91	30.41	80.61	61.78	82.59	57.60
	GEN	81.33	66.00	82.75	**62.08**	94.31	26.14	**84.61**	**59.68**	85.75	**53.47**
	GEN+ ReAct*	**82.88**	**65.64**	**82.80**	62.29	94.31	26.23	84.60	59.77	**86.15**	53.48
Require ID	KL Matching [9]	81.29	61.65	82.66	64.83	90.81	36.04	83.46	64.66	85.40	54.85
	Mahalanobis [8]	85.91	59.80	88.11	56.38	95.96	19.68	85.08	72.75	89.43	49.87
	ReAct [10]	65.42	96.29	77.68	66.45	95.13	21.93	73.95	68.39	78.04	63.27
	pNML [15]	83.23	55.37	84.19	**50.20**	92.75	28.12	83.09	61.39	85.81	48.77
	Residual [11]	83.96	59.44	86.72	59.44	92.71	31.50	84.18	73.97	88.08	52.37
	ViM [11]	**87.65**	**50.95**	**89.03**	53.28	**96.16**	**18.46**	85.28	69.81	**89.53**	**48.12**
	GEN+ ReAct [10]	86.32	56.08	84.58	59.08	94.44	25.80	84.65	**60.06**	87.50	50.26
	GEN+ Residual [11]	87.49	51.67	89.07	53.44	95.73	20.69	**85.59**	67.51	89.47	48.33

Table 4: *Average Performance of OOD Detection Methods.* Results are shown for RepVGG [41], ResNet-50-D [42], ViT [40], and DeiT [43] architectures with ImageNet-1K as ID data. The reported are averaged results over four OOD datasets: OpenImage-O, Textures, iNaturalist and ImageNet-O. For GEN, the number of maximal logits is set to 10% and $\gamma = 0.1$. Clipping quantile for ReAct* is set to 0.9995, and for ReAct [10] — to 0.999. The best performing method is in bold, the second best is underlined.

4.2 Choice of M and γ

We empirically show how the performance of our method varies with different M and γ in terms of AUROC and FPR95. First, we investigate the effective value of M for $C = 1000$ semantic classes. The first row of Figure 4 shows the results (with $\gamma = 0.1$). The results of BiT [39] are illustrated in the two rightmost columns (with AUROC and FPR95, respectively) and the results of Swin [2] are presented in the two leftmost columns (with AUROC and FPR95, respectively). It shows that it is sufficient to use the top $M = 100$ classes for

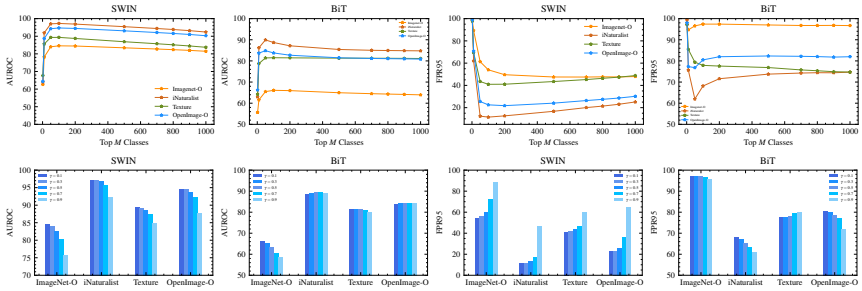

Figure 4: *Effective value of M and γ.* GEN Performance for varying values of (top row) the number of largest probabilities referred as top M classes, and (bottom row) the exponential scale γ of the entropy. The left two columns correspond to Swin [2] architecture, and the right two columns correspond to BiT [39].

the score.

We also look into the effectiveness of using different γ. The second row of Figure 4 (with $M = 100$) shows that it is adequate to obtain better OOD performance via setting $\gamma = 0.1$ for different OOD datasets. On average, AUROC and FPR95 values are better when using lower γ. Setting $\gamma = 0.1$ also works well on other architectures. Results for the remaining architectures and the dependence of (γ, M) on the architecture (which led to our choice of $(\gamma = 0.1, M = 100)$) can be found in the supplementary material.

The current evaluation protocol for OOD detection is performed on the test dataset directly, which is not suitable for real applications. We therefore evaluate the methods on completely unseen datasets, SUN [47] and Places [48][2]. GEN achieves the state-of-the-art performance with 1% and 3% margin in terms of both AUROC and FPR95 for post-hoc and ID requiring methods, respectively. The detailed results are in the supplementary material.

5 Discussion and Conclusions

In this work, we challenged ourselves to narrow the gap between simple and fast post-hoc OOD detection methods—those working on top of (nearly) black-box classifiers—and the "white-box" methods—those benefiting from

[2]We followed GradNorm [13] by taking the non-overlapping classes w.r.t. ImageNet-1k

extra information such as large and representative ID dataset with or without corresponding labels. The proposed entropy-based method GEN is as easy to implement as previous methods, and the only requirement it has is that the classifier admits class probabilities. Combining GEN with more feature-based and enhancing methods is one of the potential future directions for improvement.

We found that GEN performs best when using $\approx 10\%$ of the logits with the maximal response. Interestingly, a similar observation also applies to some other post-hoc scores (with different fractions of logits), *i.e.* that it might generally be a good idea to use only partial information coming from the largest logits. The lowest logits seem to introduce noise that might be particularly damaging for OOD detection in large-scale and fine-grained classification tasks with thousands of semantic classes. More details on our experiments can be found in the supplementary material.

6 Supplementary Material

I Experimental Details

Datasets Specifications of the datasets used in our experiments are summarized in Table 5. ImageNet-1K represents ID data, and ImageNet-O, OpenImage-O, iNaturalist, and Textures are the OOD datasets. We also provide additional results for two datasets used in the earlier work of GradNorm [13] — SUN [47] and Places [48].

Input Images An input image to BiT [39] is resized to 480×480. For ViT [40], it is resized to 384×384. And the size of input images to the remaining four architectures RepVGG [41], Swin [2], DeiT [43], and ResNet-50-D [42] is resized to 224×224.

Dataset	Class / Image Distribution	# Images
ImageNet-1K (val)[37]	predefined (ID) class list	50,000
ImageNet-O [46]	natural adversarial images	2,000
OpenImage-O [11]	natural (OOD) class distribution	17,632
iNaturalist [45]	predefined (OOD) class list	10,000
Textures [44]	predefined (OOD) class list	5,160
SUN [47]	predefined (OOD) class list	10,000
Places [48]	predefined (OOD) class list	10,000

Table 5: *Specifications of ID/OOD datasets.*

ReAct [10] vs. ReAct* Here we clarify the difference between the original ReAct [10] and our local version, ReAct*. To use the consistent notation with the main paper, $\underline{z}$ denotes the feature from the penultimate layer, b and b^* denote the clipping threshold of ReAct [10] and ReAct*, respectively. N is the number of samples in the training dataset, and m is the dimensionality of the extracted feature. ReAct [10] is defined as following,

$$\begin{aligned} &\texttt{ReAct}(\mathbf{z}; b) = \min(\mathbf{z}, b) \\ &\text{s.t.} \quad \frac{\textbf{card}\left(\{i : \mathbf{z}_{\text{train}}(i) < b\}\right)}{mN} = q, \end{aligned} \tag{C.5}$$

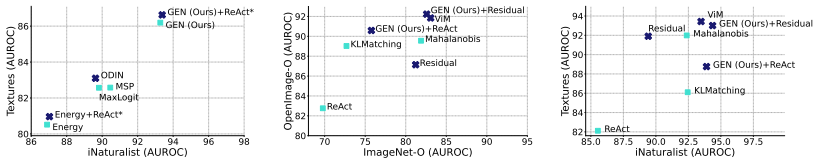

Figure 5: *Performance of (top) Post-hoc OOD Detection Methods and (mid—bottom) Methods Requiring ID Train Data Applied to 6 Classifiers Trained on ImageNet-1K.* Reported are AUROC values (%) averaged across the classifiers. Methods marked with light squares use information from logits / probabilities. Methods marked with dark crosses also use information from features.

where $\mathbf{z}_{\text{train}} = \text{flatten}(\mathbf{Z}_{\text{train}})$ is the flattened array of features $\mathbf{Z}_{\text{train}} \in \mathbb{R}^{N \times m}$ extracted from the training data, $\mathbf{card}(\cdot)$ is the cardinality and q is a predefined quantile, *e.g.*, $q = 0.99$. Intuitively, Eq. C.5 indicates that ReAct [10] employs all feature information extracted from the whole training data to find the optimal clipping threshold b. Instead, ReAct* chooses the clipping threshold based on the feature extracted from the current input only. ReAct* is defined as following,

$$\begin{aligned} &\texttt{ReAct}^*(\mathbf{z}; b) = \min(\mathbf{z}, b^*) \\ &\text{s.t.} \quad \frac{\mathbf{card}\{i : \mathbf{z}_i < b^*\}}{m} = q, \end{aligned} \tag{C.6}$$

where $\mathbf{z} \in \mathbb{R}^m$ is an output of the penultimate layer applied to the input sample.

Combining Different Scores While ViM [11] suggests employing a scaled addition of the residual and the energy score (which requires estimation of a normalization parameter), we decide to multiply the residual with our post-hoc score, GEN. This avoids the need to estimate an additional scalar parameter and appears also beneficial given the normalization property of the geometric mean.

II Averaged Performance Across Models

In Fig. 5, we report average AUROC across six classifiers for the remaining two datasets as well as the other non-post-hoc methods. One can see that GEN outperforms all the post-hoc methods on iNaturalist and Texture datasets as well as OpenImage-O and ImageNet-O shown in the main paper (see Fig. 1 of the main paper). One can also notice that GEN combined with Residual [11] is very competitive to ViM [11] on all the OOD datasets.

III Detailed OOD Detection Performance Results

We provide an extended version of Tables 3 and 4 of the main paper reporting *Per-Dataset Performance* and *Average Performance* of OOD detection methods, respectively. Due to the page capacity limitation, we split the extended results into two tables. Table 7 shows the detailed OOD detection performance on each architecture and each OOD dataset for the post-hoc methods, and Table 8 — for the methods that require ID training data. In addition, the averaged performance across all six classifiers is reported in the bottom-most block of both tables — these results are graphically visualized in Fig. 1 of the main paper and Fig. 5 in this supplementary.

Recall that we rerun the experimental evaluation of OOD detection methods according to the protocol in ViM [11] with the exception of ODIN [14], and we obtained slightly better results than reported in ViM [11]. For ODIN [14], both the code and tuned hyperparameters (scale of the perturbation ε and temperature T) were not provided by ViM, therefore its results were taken from ViM [11] paper.

IV Extended Results for Effective Value of M and γ

This section contains a more detailed evaluation for our GEN score using varying choices for M and γ. In particular, we illustrate the results for the four remaining architectures RepVGG [41], ViT [40], DeiT [43], and ResNet-50-D [42]. The results for varying $M \in \{2, 10, 50, 100, 200, 500, 700, 800, 900, 1000\}$ are depicted in Fig. 6, where it can be seen that using more logit information causes OOD detection performance to degrade for most architectures except for ViT [40]. Besides, setting $M = 100$ seems perform well in terms of AUROC

and FPR95 generally. The results of using different $\gamma = \{0.1, 0.3, 0.5, 0.9\}$ are shown in Fig. 7. The top row shows that using larger γ barely improves the performance in terms of AUROC. The same observation can be made regarding FPR95, which is shown in the bottom row.

V Performance on Unseen Datasets

We perform OOD detection on two completely unseen OOD datasets from SUN [47] and Places [48]. Importantly, the overlapped classes between SUN / Places and ImageNet-1K are removed as provided by [13]. We use the previously validated hyperparameters $M = 100$ and $\gamma = 0.1$. The results can be found in Table 6 indicating a consistently better performance of GEN.

VI Using the Top Logits for the Energy Score

We empirically verify the hypothesis that using only the partial information from the largest logits is beneficial. In particular, the smallest logits seem to introduce noise that might be especially detrimental for OOD detection in large scale and fine-grained classification tasks with a large number of semantic classes. The main paper has a respective evaluation for our proposed score w.r.t Swin [2] and BiT [39] architectures (see Fig. 4 in

OOD Method	**SUN**		**Places**		**Average**	
	A↑	F↓	A↑	F↓	A↑	F↓
Averaged						
MSP [7]	<u>83.97</u>	64.39	<u>82.18</u>	69.48	<u>83.08</u>	66.93
MaxLogit [9]	81.86	<u>62.34</u>	79.48	<u>67.38</u>	80.67	<u>64.86</u>
Energy [1]	79.53	65.13	76.68	70.72	78.11	67.93
GradNorm [13]	54.91	78.64	51.34	83.65	53.13	81.14
GEN (Ours)	**84.99**	**61.34**	**82.79**	**65.98**	**83.89**	**63.66**
KL Matching [9]	82.76	69.70	81.26	72.20	82.01	70.95
Mahalanobis [8]	81.88	72.25	79.40	75.36	80.64	73.81
ReAct [10]	77.61	65.08	74.25	71.42	75.93	68.25
pNML [15]	84.46	<u>58.23</u>	82.05	<u>64.90</u>	83.26	<u>61.57</u>
Residual [11]	78.53	77.66	75.52	80.40	77.03	79.03
ViM [11]	<u>84.93</u>	64.97	<u>82.06</u>	69.45	<u>83.50</u>	67.21
GEN (Ours) + Residual [11]	**88.54**	**52.37**	**84.79**	**64.05**	**86.67**	**58.21**

Table 6: *OOD Detection Performance on Unseen Datasets.*

Architecture + OOD Method	OpenImage-O		Textures		iNaturalist		ImageNet-O		Average	
	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓
BiT-S-R101x1										
MSP [7]	83.05	76.21	79.76	77.13	87.90	64.53	57.16	96.90	76.97	78.69
MaxLogit [9]	82.33	79.75	81.65	73.59	86.78	70.52	62.99	96.90	78.44	80.19
Energy [1]	80.59	82.00	81.10	73.91	84.52	74.93	63.56	96.35	77.44	81.80
GradNorm [13]	70.68	79.34	**83.12**	**55.72**	86.13	**58.34**	53.73	**91.90**	73.42	**71.33**
ODIN [14]	**85.64**	**72.83**	81.60	74.07	86.73	70.75	63.00	96.85	79.24	78.63
ReAct*	80.83	81.85	81.44	73.74	84.77	74.80	63.63	96.30	77.67	81.67
Shannon Entropy	83.98	80.48	81.30	76.32	88.73	69.66	60.42	97.30	78.61	80.94
GEN (Ours)	83.77	80.43	81.48	77.93	88.67	68.32	66.09	97.30	80.00	81.00
GEN (Ours) + ReAct*	83.99	80.35	81.80	77.87	**88.90**	68.03	**66.18**	97.25	**80.22**	80.88
DeiT										
MSP [7]	83.85	61.65	81.98	64.46	88.27	52.02	63.66	86.75	79.44	66.22
MaxLogit [9]	80.01	60.44	80.42	61.10	85.24	52.60	61.40	83.35	76.77	64.37
Energy [1]	74.56	66.36	77.41	64.77	78.64	65.80	60.63	**82.60**	72.81	69.88
GradNorm [13]	27.63	97.96	38.96	94.75	28.56	98.90	33.06	98.25	32.05	97.47
ODIN [14]	80.19	59.53	81.26	59.38	85.36	51.81	61.70	84.95	77.13	63.92
ReAct*	74.57	66.35	77.42	64.81	78.67	65.62	60.62	82.70	72.82	69.87
Shannon Entropy	84.71	57.54	83.50	59.05	89.29	47.55	64.93	83.00	80.61	61.78
GEN (Ours)	**88.34**	**55.63**	**86.49**	**56.36**	**92.29**	**42.52**	**71.33**	84.20	**84.61**	**59.68**
GEN (Ours) + ReAct*	88.33	55.72	86.48	56.45	92.27	42.68	**71.33**	84.25	84.60	59.77
RepVGG										
MSP [7]	84.72	64.04	78.58	72.69	87.10	55.02	61.67	91.55	78.02	70.83
MaxLogit [9]	84.48	65.45	76.31	76.71	86.21	62.15	62.89	89.90	77.47	73.55
Energy [1]	83.36	70.08	74.51	82.87	83.92	75.49	63.38	88.00	76.29	79.11
GradNorm [13]	52.48	94.81	58.25	91.30	53.40	98.20	47.79	95.60	52.98	94.98
ODIN [14]	85.22	63.48	76.77	76.14	86.37	61.40	62.50	89.70	77.72	72.68
ReAct*	84.66	69.23	76.39	82.46	84.30	74.84	65.05	**87.75**	77.60	78.57
Shannon Entropy	85.82	64.09	78.86	74.92	87.77	58.55	63.60	89.70	79.01	71.81
GEN (Ours)	87.46	59.86	80.98	67.42	90.56	45.32	66.33	91.40	81.33	66.00
GEN (Ours) + ReAct*	**88.66**	**59.31**	**83.20**	**67.07**	**91.00**	**44.78**	**68.67**	91.40	**82.88**	**65.64**
ResNet-50-D										
MSP [7]	84.56	63.55	82.71	64.71	88.57	50.38	56.14	93.75	77.99	68.10
MaxLogit [9]	81.90	65.04	79.17	66.16	86.39	53.35	54.40	92.55	75.47	69.28
Energy [1]	76.72	75.07	73.85	75.48	80.44	71.54	53.99	89.95	71.25	78.01
GradNorm [13]	38.85	97.75	54.68	90.41	41.74	98.06	40.88	98.10	44.04	96.08
ODIN [14]	81.53	64.49	80.21	63.93	86.48	52.58	52.87	93.25	75.27	68.56
ReAct*	77.01	74.88	74.32	75.12	80.59	70.94	54.27	**89.85**	71.55	77.70
Shannon Entropy	85.12	62.40	83.18	62.77	89.23	48.67	57.75	91.80	78.82	66.41
GEN (Ours)	88.09	**58.59**	86.43	**57.25**	**92.25**	**39.97**	64.24	92.50	82.75	**62.08**
GEN (Ours) + ReAct*	**88.14**	58.82	**86.50**	57.48	92.23	40.36	**64.34**	92.50	**82.80**	62.29
Swin										
MSP [7]	91.38	34.81	85.31	51.74	94.76	22.97	78.86	63.90	87.58	43.36
MaxLogit [9]	92.09	26.70	84.81	47.23	95.71	15.34	81.07	52.10	88.42	35.34
Energy [1]	91.24	26.92	82.80	51.57	95.19	15.49	82.00	**45.85**	87.81	34.96
GradNorm [13]	45.52	77.94	37.12	93.02	33.79	88.81	50.27	78.05	41.68	84.45
ODIN [14]	91.38	28.42	85.74	44.59	94.24	19.65	80.62	53.65	88.00	36.58
ReAct*	91.23	26.98	82.79	51.69	95.18	15.50	82.00	45.90	87.80	35.02
Shannon Entropy	93.16	25.61	87.15	43.84	95.95	16.21	82.13	51.95	89.60	34.40
GEN (Ours)	**94.70**	**22.60**	**89.43**	**40.95**	**97.25**	**11.55**	**84.45**	54.00	**91.46**	**32.28**
GEN (Ours) + ReAct*	94.69	22.62	89.42	41.01	**97.25**	11.56	84.44	54.00	91.45	32.30
ViT-B/16										
MSP [7]	92.17	34.96	87.13	48.45	96.13	19.14	81.88	65.00	89.33	41.89
MaxLogit [9]	96.73	16.58	93.05	30.27	98.57	6.53	89.88	44.00	94.56	24.34
Energy [1]	**96.99**	**14.78**	**93.42**	**28.14**	**98.66**	6.04	**90.49**	**41.20**	**94.89**	**22.54**
GradNorm [13]	93.79	20.94	89.76	34.26	97.34	8.54	80.38	50.90	90.32	28.66
ODIN [14]	96.86	15.68	93.01	30.60	98.57	6.58	89.85	44.15	94.57	24.25
ReAct*	96.98	14.87	93.41	28.35	**98.66**	6.01	**90.49**	42.10	**94.89**	22.83
Shannon Entropy	94.81	22.24	89.82	38.18	97.92	8.71	85.10	52.50	91.91	30.41
GEN (Ours)	96.60	17.13	92.35	34.01	98.63	**5.83**	89.67	47.60	94.31	26.14
GEN (Ours) + ReAct*	96.60	17.19	92.35	34.07	98.63	5.85	89.67	47.80	94.31	26.23
Averaged										
MSP [7]	86.62	55.87	82.58	63.20	90.45	44.01	66.56	82.97	81.55	61.51
MaxLogit [9]	86.26	52.33	82.57	59.18	89.82	43.41	68.77	76.47	81.85	57.85
Energy [1]	83.91	55.87	80.52	62.79	86.89	51.55	69.01	**73.99**	80.08	61.05
GradNorm [13]	54.82	78.12	60.31	76.58	56.83	75.14	51.02	85.47	55.75	78.83
ODIN [14]	86.80	50.74	83.10	58.12	89.62	43.79	68.42	77.09	81.98	57.44
ReAct*	84.21	55.69	80.96	62.70	87.03	51.29	69.34	74.10	80.39	60.94
Shannon Entropy	81.98	52.06	83.97	59.18	91.48	41.56	68.99	70.71	83.09	57.63
GEN (Ours)	89.83	49.04	86.19	**55.65**	93.27	35.59	73.69	77.83	85.74	54.53
GEN (Ours) + ReAct*	**90.07**	**49.00**	**86.62**	55.66	**93.38**	**35.54**	**74.11**	77.87	**86.04**	**54.52**

Table 7: *Performance of Post-hoc Methods.* *BiT-S-R101x1*, *DeiT*, *RepVGG*, *ResNet-50-D*, *Swin*, and *ViT-B/16*are included along with the averaged performance across models. The ID dataset is ImageNet-1K, the OOD datasets are OpenImage-O, Textures, iNaturalist and ImageNet-O. Units for AUROC and FPR95 are percentages. The best performing method is in bold, the second best is underlined.

Architecture + OOD Method	OpenImage-O		Textures		iNaturalist		ImageNet-O		Average	
	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓
BiT-S-R101x1										
KL Matching [9]	87.94	54.92	86.91	50.89	92.95	**33.19**	65.76	86.80	83.39	56.45
Mahalanobis [8]	82.62	66.24	97.33	13.95	85.79	64.71	80.37	70.20	86.53	53.77
ReAct [10]	82.54	79.06	84.85	68.60	86.89	70.16	64.81	95.65	79.77	78.37
pNML [15]	88.62	55.27	93.59	22.25	**93.12**	38.21	67.27	86.35	85.65	50.52
Residual [11]	80.20	68.05	97.67	11.14	76.93	80.18	81.58	65.60	84.09	56.24
ViM [11]	89.96	49.01	**98.92**	**4.63**	89.38	55.09	83.85	**61.25**	90.53	42.50
GEN (Ours) + ReAct [10]	87.44	70.07	88.35	63.86	91.63	53.30	70.81	93.80	84.56	70.26
GEN (Ours) + Residual [11]	**91.75**	**43.83**	98.54	5.78	92.25	47.13	**83.88**	63.70	**91.61**	**40.11**
DeiT										
KL Matching [9]	87.29	60.58	84.88	63.35	90.56	50.45	71.09	84.25	83.46	64.66
Mahalanobis [8]	89.18	64.84	83.60	77.13	91.55	58.78	**75.98**	90.25	85.08	72.75
ReAct [10]	75.95	64.80	78.03	64.28	80.63	62.22	61.17	**82.25**	73.95	68.39
pNML [15]	86.68	57.86	86.02	**56.32**	90.54	47.45	69.10	83.95	83.09	61.39
Residual [11]	88.16	68.56	82.70	77.58	91.30	58.45	74.58	91.30	84.19	73.97
ViM [11]	89.21	63.84	84.43	73.12	92.13	52.86	75.34	89.40	85.28	69.81
GEN (Ours) + ReAct [10]	88.43	**55.84**	**86.46**	56.90	92.30	**43.06**	71.42	84.45	84.65	**60.06**
GEN (Ours) + Residual [11]	**89.46**	61.96	84.90	70.04	**92.58**	49.05	75.42	89.00	**85.59**	67.51
RepVGG										
KL Matching [9]	86.49	57.53	83.20	61.92	89.06	42.24	66.42	84.90	81.29	61.65
Mahalanobis [8]	85.16	66.18	92.69	32.13	89.14	58.92	76.65	81.95	85.91	59.80
ReAct [10]	67.37	96.93	68.25	94.13	66.25	99.19	59.79	94.90	65.42	96.29
pNML [15]	88.75	**49.92**	86.02	44.22	89.91	46.67	68.23	80.65	83.23	55.37
Residual [11]	81.70	66.73	93.03	28.66	86.05	62.45	75.06	79.90	83.96	59.44
ViM [11]	88.68	53.82	**93.68**	**23.88**	91.33	46.91	**76.90**	**79.20**	**87.65**	**50.95**
GEN (Ours) + ReAct [10]	**88.99**	52.85	90.35	48.82	91.82	**36.76**	74.13	85.90	86.32	56.08
GEN (Ours) + Residual [11]	**88.99**	53.89	92.73	28.00	**92.16**	42.80	76.09	82.00	87.49	51.67
ResNet-50-D										
KL Matching [9]	87.13	60.88	86.06	61.92	90.48	47.66	66.96	88.85	82.66	64.83
Mahalanobis [8]	88.69	58.71	94.15	28.14	89.51	62.34	80.10	76.35	88.11	56.38
ReAct [10]	81.63	66.16	84.68	54.17	84.55	60.71	59.86	84.75	77.68	66.45
pNML [15]	88.72	**47.86**	91.28	32.62	91.36	39.53	65.39	80.80	84.19	**50.20**
Residual [11]	86.47	62.86	94.63	25.66	84.70	75.79	**81.10**	**73.45**	86.72	59.44
ViM [11]	90.00	53.50	**95.84**	**20.48**	89.29	64.43	80.98	74.70	89.03	53.28
GEN (Ours) + ReAct [10]	89.20	55.86	89.17	50.93	**92.72**	**38.48**	67.24	91.05	84.58	59.08
GEN (Ours) + Residual [11]	**90.18**	53.41	95.24	23.51	90.67	58.33	80.19	78.50	**89.07**	53.44
Swin										
KL Matching [9]	91.86	39.93	86.82	53.24	94.75	27.76	81.78	67.30	88.80	47.06
Mahalanobis [8]	94.35	34.85	89.95	49.09	98.69	5.38	85.43	73.65	92.11	40.74
ReAct [10]	91.83	25.92	83.33	50.54	95.90	13.84	82.26	45.75	88.33	34.01
pNML [15]	95.53	**19.29**	91.55	**33.29**	97.84	8.98	87.22	**45.05**	93.03	**26.65**
Residual [11]	94.44	33.40	91.36	43.26	98.90	4.79	86.66	68.65	92.84	37.53
ViM [11]	**95.93**	24.43	**92.40**	37.98	**99.29**	**2.62**	**88.74**	59.00	**94.09**	31.01
GEN (Ours) + ReAct [10]	95.09	21.94	89.71	41.22	97.75	9.45	84.84	56.10	91.85	32.18
GEN (Ours) + Residual [11]	95.73	25.06	92.23	37.66	99.13	3.10	88.07	61.50	93.79	31.83
ViT-B/16										
KL Matching [9]	93.46	29.58	88.75	43.84	96.88	15.03	84.14	55.70	90.81	36.04
Mahalanobis [8]	**97.33**	14.32	94.21	25.27	**99.53**	**2.15**	**92.78**	37.00	95.96	19.69
ReAct [10]	97.24	**13.99**	93.54	27.62	99.01	4.21	90.74	41.90	95.13	21.93
pNML [15]	95.38	20.33	90.98	34.53	98.18	7.69	86.44	49.95	92.75	28.12
Residual [11]	91.86	36.41	92.04	34.73	98.58	6.56	88.35	48.30	92.71	31.50
ViM [11]	97.30	14.39	**95.31**	**20.14**	99.41	2.56	92.61	**36.75**	**96.16**	**18.46**
GEN (Ours) + ReAct [10]	96.77	16.37	92.41	33.70	98.95	4.34	89.79	47.95	94.48	25.59
GEN (Ours) + Residual [11]	97.29	14.17	94.41	25.17	99.38	2.67	91.83	40.75	95.73	20.69
Averaged										
KL Matching [9]	89.03	50.57	86.10	55.86	92.45	36.05	72.69	77.97	85.07	55.11
Mahalanobis [8]	89.56	50.86	91.99	37.62	92.37	42.05	81.89	71.57	88.95	50.52
ReAct [10]	82.76	57.81	82.11	59.89	85.54	51.72	69.77	74.20	80.05	60.91
pNML [15]	90.61	**41.76**	89.91	37.20	93.49	**31.42**	73.94	71.12	86.99	45.38
Residual [11]	87.14	56.00	91.90	36.84	89.41	48.04	81.22	71.20	87.42	53.02
ViM [11]	91.85	43.16	**93.43**	**30.04**	93.47	37.41	**83.07**	**66.72**	90.45	44.33
GEN (Ours) + ReAct [10]	90.59	46.94	88.76	50.91	93.89	32.70	75.76	76.76	87.25	51.83
GEN (Ours) + Residual [11]	**92.23**	42.05	93.01	31.69	**94.36**	33.85	82.58	69.24	**90.55**	**44.21**

Table 8: *Performance of Methods Requiring ID Data. BiT-S-R101x1*, *DeiT*, *RepVGG*, *ResNet-50-D*, *Swin*, and *ViT-B/16*are included along with the averaged performance across models. The ID dataset is ImageNet-1K, the OOD datasets are OpenImage-O, Textures, iNaturalist and ImageNet-O. Units for AUROC and FPR95 are percentages. The best performing method is in bold, the second best is underlined.

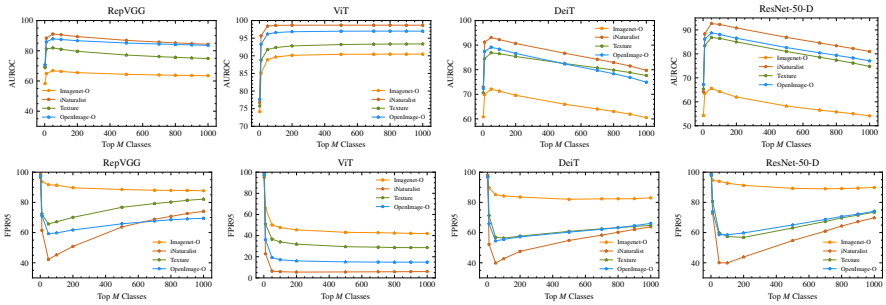

Figure 6: *OOD Detection Performance of GEN Score with Varying M.* Reported are (top) AUROC and (bottom) FPR95 values.

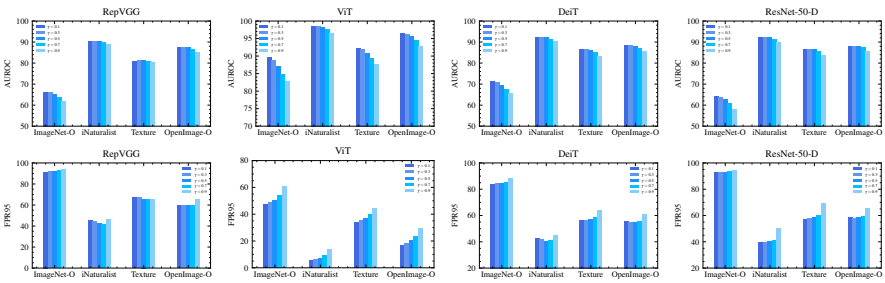

Figure 7: *OOD Detection Performance of GEN Score with Varying γ.* Reported are (top) AUROC and (bottom) FPR95 values.

the main paper), and here we demonstrate a similar behavior for the Energy [1] score. The original Energy [1] method simply uses all logits to calculate the score. We instead utilize a subset of M largest logits. The results for $M = \{1, 2, 5, 10, 20, 30, 50, 100, 200, 500, 700, 1000\}$ are shown in Fig. 8. It shows that AUROC decreases and FPR95 increases for most of the classifiers except for ViT [40] when the number of incorporated logits is increased. That is to say, using more logits indeed degrades the OOD detection performance of most architectures (with the exception of ViT [40]).

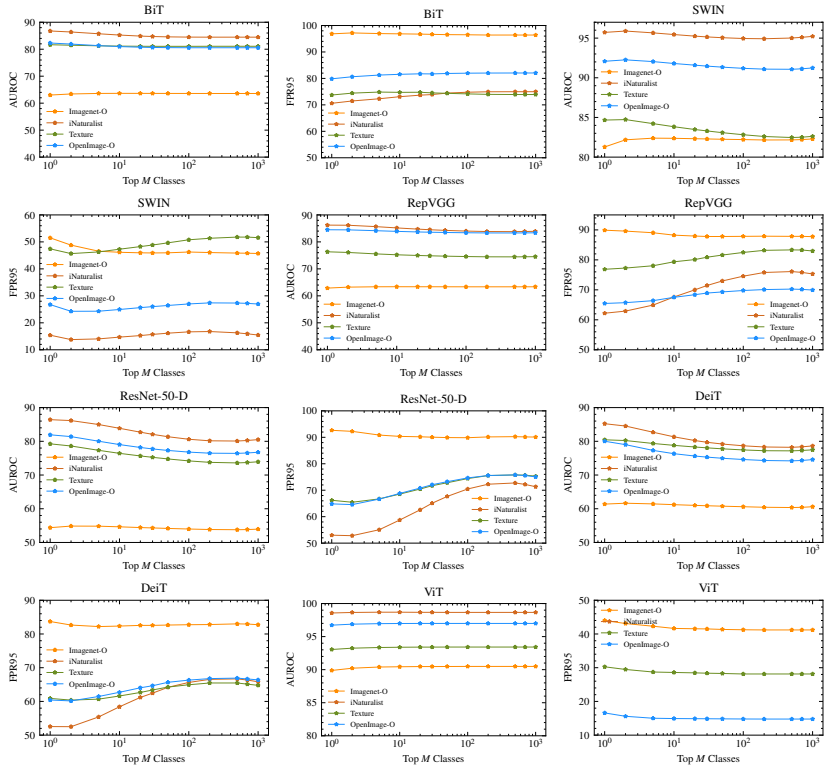

Figure 8: *OOD Detection Performance of Energy Score with Varying M.* Reported are (left) AUROC and (right) FPR95 values.

VII Sensitivity to Temperature Scaling

A pretrained network might also be adjusted to yield better calibrated predictions. Since calibration methods rely on some training data, which cannot be assumed to be available, we investigate into the sensitivity of post-hoc OOD scores w.r.t. applying a classifier calibration. In particular, we simulate the effects of the simple and popular temperature scaling approach [24], which scales the logits by an inverse temperature $1/T$. Once the right temperature T is determined (using validation data), it can be absorbed into the layer generated the logits (and therefore the original logits might become inaccessible). We simulate temperatures $T \in \{0.2, 0.5, 1, 2, 5\}$ and illustrate the sensitivity

of AUROC and FPR95 values for post-hoc OOD detection scores in Table 9.

The MaxLogit [9] score is agnostic to temperature scaling by construction. It can be seen that GEN is relatively insensitive to temperature scaling in terms of AUROC values, but shows some sensitivity in the FPR95 results. Energy [1] is slightly less sensitive than GEN in terms of FPR95 score, but more sensitive in terms of AUROC score, and MSP [7] overall is more sensitive. GradNorm [13] shows the highest sensitivity to temperature scaling. Note that all methods (except the invariant MaxLogit [9] score) are relatively sensitive in their FPR95 results.

Method	Average Performance	
	AUROC ↑	FPR95 ↓
MSP [7] Temp-0.2	77.99	78.75
MSP [7] Temp-0.5	81.85	69.41
MSP [7] Temp-1	87.58	43.36
MSP [7] Temp-2	**88.51**	37.33
MSP [7] Temp-5	88.03	**37.24**
MaxLogit [9] Temp-0.2	88.42	35.34
MaxLogit [9] Temp-0.5	88.42	35.34
MaxLogit [9] Temp-1	88.42	35.34
MaxLogit [9] Temp-2	88.42	35.34
MaxLogit [9] Temp-5	88.42	35.34
Energy [1] Temp-0.2	88.48	35.03
Energy [1] Temp-0.5	**88.70**	**33.82**
Energy [1] Temp-1	87.81	34.96
Energy [1] Temp-2	62.34	61.23
Energy [1] Temp-5	62.43	61.67
GradNorm [13] Temp-0.2	13.47	99.84
GradNorm [13] Temp-0.5	15.70	98.89
GradNorm [13] Temp-1	**41.68**	**84.45**
GradNorm [13] Temp-2	19.25	99.50
GradNorm [13] Temp-5	14.37	99.85
GEN (Ours) Temp-0.2	89.53	38.94
GEN (Ours) Temp-0.5	90.82	34.74
GEN (Ours) Temp-1	**91.46**	**32.27**
GEN (Ours) Temp-2	87.24	61.46
GEN (Ours) Temp-5	84.23	69.88

Table 9: *Sensitivity to Temperature Scaling.* The reported is the average performance across 6 classifiers —*BiT-S-R101x1*, *DeiT*, *RepVGG*, *ResNet-50-D*, *Swin*, and *ViT-B/16*— and 4 datasets — OpenImage-O, Textures, iNaturalist, and ImageNet-O.

ARCH + OOD Method	**iNaturalist**		**Texture**		**OpenImage-O**		**ImageNet-O**	
	A↑	F↓	A↑	F↓	A↑	F↓	A↑	F↓
BiT-S-R101x1								
FeatureNorm	74.67	77.50	74.30	65.95	53.97	87.64	50.54	93.30
ProbsDistance	86.66	73.96	81.27	77.05	82.51	82.49	65.64	96.95
GradNorm [13]	86.13	**58.34**	**83.12**	**55.72**	70.68	79.34	53.73	**91.90**
GEN (Ours)	**88.67**	68.32	81.48	77.93	**83.77**	80.43	**66.09**	97.30
Swin								
FeatureNorm	4.05	100.00	15.65	99.61	11.32	99.90	22.55	99.90
ProbsDistance	94.64	20.78	86.33	45.43	92.45	26.78	82.91	**47.85**
GradNorm [13]	33.79	88.81	37.12	93.02	45.52	77.94	50.27	78.05
GEN (Ours)	**97.25**	**11.55**	**89.43**	**40.95**	**94.70**	**22.60**	**84.45**	54.00

Table 10: *Feature vs. Probability Space.* Using feature norms in most cases degrades the performance hence making GradNorm [13] unstable especially on the largest OpenImage-O[38] dataset.

VIII Comparison with GradNorm [13]

We conduct extra experiments on BiT [39] and Swin [2] to test our approach. First, we compare our method to the recent post-hoc method GradNorm [13], which claims that using joint information from feature space and probability space is helpful for OOD detection. Based on our experimental observations, it is not always true, and the performance depends on the model architecture. Second,

It is claimed in GradNorm [13] that using joint information from feature space and probability space could achieve better OOD results. There, feature information is represented as feature norm $\|\mathbf{z}\|_1$, and probability information is compressed as the total variation (*i.e.* l_1-distance) between uniform distribution and predictive distribution $\|\mathbf{p} - \mathbf{u}\|_1$. We further investigate whether this conclusion holds for other architectures and OOD datasets. We reproduce and extend Table 5 of GradNorm [13] for all six architectures and six OOD datasets. The results for BiT [39] and Swin [2] are shown in Table 10, and results for other architectures can be found in supplementary material. It can be seen that feature norms $\|\mathbf{z}\|_1$ are not always distinctive for OOD detection and could cause occasional bad performance of GradNorm [13]. Besides, our score which only uses information from probability space outperforms the score using probability distance and GradNorm [13] in most datasets.

IX Analysis of GradNorm [13]: Dependence on the Checkpoint

We compare the performance of OOD detection methods for BiT-S-R101x1 architecture with two different weights (checkpoints). The first one is the official checkpoint of BiT [39] used by ViM [11], and the second one is the fine-tuned set of weights provided by GradNorm [13]. The results in Table 11 are averaged AUROC and FPR95 on four OOD datasets. One can notice that GradNorm [13] performs worse when official checkpoint is used. However the downstream performance—ImageNet classification, see Table 2 in the main paper—is worse for the fine-tuned checkpoint from GradNorm [13] indicating a certain bias in GradNorm [13] checkpoint. Moreover, GEN consistently outperforms GradNorm [13] on the two most challenging datasets, OpenImage-O and ImageNet-O.

Arch + Method	**iNaturalist**		**Texture**		**OpenImage-O**		**ImageNet-O**	
	A↑	F↓	A↑	F↓	A↑	F↓	A↑	F↓
BiT-S-R101x1								
MSP [7]	87.90	64.53	79.76	77.1	83.05	**76.21**	57.16	96.90
MaxLogit [9]	86.78	70.52	81.65	73.59	82.33	79.75	62.99	96.90
GradNorm [13]	86.13	**58.34**	**83.12**	**55.72**	70.68	79.34	53.73	**91.90**
GEN (Ours)	**88.67**	68.32	81.48	77.93	**83.77**	80.43	**66.09**	97.30
BiT-S-R101x1 [13]								
MSP [7]	87.57	63.94	76.87	81.51	80.18	80.56	55.55	97.65
MaxLogit [9]	89.38	62.71	78.53	79.81	80.35	81.93	59.26	97.70
GradNorm [13]	**90.45**	**49.41**	**83.30**	**58.74**	73.59	**79.13**	54.43	**93.45**
GEN (Ours)	89.03	68.20	77.85	86.45	**81.34**	83.31	**63.26**	97.45

Table 11: *OOD Detection Performance Depends on Checkpoint.* OOD detection results for BiT-S-R101x1 with official checkpoint [39] and the one provided by GradNorm [13]. The performance of GradNorm [13] gets worse for the official weights.

References

[1] W. Liu, X. Wang, J. Owens, and Y. Li, "Energy-based out-of-distribution detection," in *Advances in Neural Information Processing Systems*, 2020.

[2] Z. Liu, Y. Lin, Y. Cao, *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[3] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan, "Do deep generative models know what they don't know?" In *International Conference on Learning Representations (ICLR)*, 2019.

[4] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[5] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *Information Processing in Medical Imaging (IPMI)*, 2017.

[6] J. Yang, K. Zhou, Y. Li, and Z. Liu, "Generalized out-of-distribution detection: A survey," *arXiv:2110.11334*, 2021.

[7] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[8] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *Advances in Neural Information Processing Systems*, 2018.

[9] D. Hendrycks, S. Basart, M. Mazeika, *et al.*, "Scaling out-of-distribution detection for real-world settings," in *International Conference on Machine Learning (ICML)*, 2022.

[10] Y. Sun, C. Guo, and Y. Li, "React: Out-of-distribution detection with rectified activations," in *Advances in Neural Information Processing Systems*, 2021.

[11] H. Wang, Z. Li, L. Feng, and W. Zhang, "Vim: Out-of-distribution with virtual-logit matching," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[12] Y. Song, N. Sebe, and W. Wang, "Rankfeat: Rank-1 feature removal for out-of-distribution detection," in *Advances in Neural Information Processing Systems*, 2022.

[13] R. Huang, A. Geng, and Y. Li, "On the importance of gradients for detecting distributional shifts in the wild," in *Advances in Neural Information Processing Systems*, 2021.

[14] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *International Conference on Learning Representations (ICLR)*, 2018.

[15] K. Bibas, M. Feder, and T. Hassner, "Single layer predictive normalized maximum likelihood for out-of-distribution detection," in *Advances in Neural Information Processing Systems*, 2021.

[16] W. Grathwohl, K.-C. Wang, J.-H. Jacobsen, D. Duvenaud, M. Norouzi, and K. Swersky, "Your classifier is secretly an energy based model and you should treat it like one," in *International Conference on Learning Representations (ICLR)*, 2020.

[17] Y.-C. Hsu, Y. Shen, H. Jin, and Z. Kira, "Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[18] T. DeVries and G. W. Taylor, "Learning confidence for out-of-distribution detection in neural networks," *arXiv:1802.04865*, 2018.

[19] A. Zaeemzadeh, N. Bisagno, Z. Sambugaro, N. Conci, N. Rahnavard, and M. Shah, "Out-of-distribution detection using union of 1-dimensional subspaces," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[20] R. Huang and Y. Li, "Mos: Towards scaling out-of-distribution detection for large semantic space," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[21] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," in *International Conference on Learning Representations (ICLR)*, 2019.

[22] A.-A. Papadopoulos, M. R. Rajati, N. Shaikh, and J. Wang, "Outlier exposure with confidence control for out-of-distribution detection," *Neurocomputing*, vol. 441, pp. 138–150, 2021.

[23] T. Silva Filho, H. Song, M. Perello-Nieto, R. Santos-Rodriguez, M. Kull, and P. Flach, "Classifier calibration: How to assess and improve predicted class probabilities: A survey," *arXiv:2112.10327*, 2021.

[24] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[25] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates," in *ACM SIGKDD*, 2002.

[26] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International Conference on Machine Learning (ICML)*, 2017.

[27] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems*, 2017.

[28] V. Papyan, X. Han, and D. L. Donoho, "Prevalence of neural collapse during the terminal phase of deep learning training," *the National Academy of Sciences*, vol. 117, no. 40, pp. 24 652–24 663, 2020.

[29] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[30] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[31] F. Liese and I. Vajda, "On divergences and informations in statistics and information theory," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4394–4412, 2006.

[32] L. V. Kantorovich, "Mathematical methods of organizing and planning production," *Management science*, 1960.

[33] C. Villani, "Optimal transport: Old and new," in *Springer*, vol. 338, 2009.

[34] T. Gneiting and A. E. Raftery, "Strictly proper scoring rules, prediction, and estimation," *Journal of the American statistical Association*, 2007.

[35] A. P. Dawid and M. Musio, "Theory and applications of proper scoring rules," *Metron*, 2014.

[36] A. Buja, W. Stuetzle, and Y. Shen, "Loss functions for binary class probability estimation and classification: Structure and applications," *Working draft, November*, 2005.

[37] O. Russakovsky, J. Deng, H. Su, *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, 2015.

[38] I. Krasin, T. Duerig, N. Alldrin, *et al.*, "Openimages: A public dataset for large-scale multi-label and multi-class image classification.," *Dataset available from https://github.com/openimages*, 2017.

[39] A. Kolesnikov, L. Beyer, X. Zhai, *et al.*, "Big transfer (bit): General visual representation learning," in *European Conference on Computer Vision (ECCV)*, 2020.

[40] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations (ICLR)*, 2021.

[41] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "Repvgg: Making vgg-style convnets great again," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[42] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[43] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, "Training data-efficient image transformers & distillation through attention," in *International Conference on Machine Learning (ICML)*, 2021.

[44] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[45] G. Van Horn, O. Mac Aodha, Y. Song, *et al.*, "The inaturalist species classification and detection dataset," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[46] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, "Natural adversarial examples," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[47] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[48] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

PAPER D

Deep Nearest Neighbors for Anomaly Detection in Chest X-Rays

Xixi Liu, Jennifer Alvén, Ida Häggström, Christopher Zach

International Workshop on Machine Learning in Medical Imaging (MIML), held in conjunction with MICCAI
pp. 293–302, 2023
DOI: 10.1007/978-3-031-45676-3-30

The layout has been revised.

Abstract

[a]Identifying medically abnormal images is crucial to the diagnosis procedure in medical imaging. Due to the scarcity of annotated abnormal images, most reconstruction-based approaches for anomaly detection are trained only with normal images. At test time, images with large reconstruction errors are declared abnormal. In this work, we propose a novel feature-based method for anomaly detection in chest x-rays in a setting where only normal images are provided during training. The model consists of lightweight adaptor and predictor networks on top of a pre-trained feature extractor. The parameters of the pre-trained feature extractor are frozen, and training only involves fine-tuning the proposed adaptor and predictor layers using Siamese representation learning. During inference, multiple augmentations are applied to the test image, and our proposed anomaly score is simply the geometric mean of the k-nearest neighbor distances between the augmented test image features and the training image features. Our method achieves state-of-the-art results on two challenging benchmark datasets, the RSNA Pneumonia Detection Challenge dataset, and the VinBigData Chest X-ray Abnormalities Detection dataset. Furthermore, we empirically show that our method is robust to different amounts of anomalies among the normal images in the training dataset. The code is available at: https://github.com/XixiLiu95/deep-kNN-anomaly-detection.

[a]This work is partially supported by the Wallenberg Artificial Intelligence, Autonomous Systems and Software Program (WASP) funded by Knut and Alice Wallenberg Foundation.

1 Introduction

Chest X-rays (CXRs) are commonly considered the main imaging study for the evaluation of many conditions because of their cost-effectiveness, low radiation dose, and versatility as a diagnostic tool [1]. Deep learning image

analysis methods, with fast inference and high accuracy, can help improve the efficiency of image evaluation and the diagnostic accuracy as well as reduce the workload for radiologists [2]. However, for such methods to be safe and reliable, we need robust methods for detecting anomalies in the input data. Consequently, anomaly detetcion has been extensively studied and is an important sub-routine in many computer-aided diagnosis methods [2]. A recent paper on anomaly detection in medical images summarizes several scenarios requiring anomaly detection including but not limited to rejecting inputs that are incorrectly prepared (e.g. blurry images, poor contrast, and incorrect view) and rejecting inputs that are unseen in the training data (e.g. images with unseen diseases) [3]. In this work, we focus on the second case, where the goal is to identify medically abnormal images. However, the proposed method is general, and can with ease be applied to other scenarios (e.g. image artifacts, unusual anatomies, and artificial implants).

Anomaly detection

refers to the task of distinguishing abnormal data from normal data. In this study, our focus is on the scenario where only healthy images are accessible during the training phase. The existing methods can be roughly divided into two categories including reconstruction-based methods and self-supervised learning-based methods. Reconstruction-based methods assume that normal samples tend to produce lower reconstruction errors compared to abnormal samples. Several reconstruction-based methods are devised for anomaly detection, e.g. autoencoders (AEs) and their variants [4], [5], and generative adversarial networks (GANs) such as f-AnoGAN [6]. Recently, diffusion models and their variants have gotten attention due to their powerful mode coverage over GANs [7], [8] and due to the more realistic sample quality compared to variational autoencoders (VAEs). Most reconstruction-based methods rely on large amounts of normal training data, however, the authors in [9] argue that a large amount of unlabeled data containing outliers could be beneficial when learning anomaly detection. Their reconstruction-based dual distribution anomaly detection (DDAD) method utilizes the unlabeled data to learn the inter-discrepancy of two modules, where one is accessible to normal healthy images, and the other is accessible to the unlabeled images. While the DDAD method achieves impressive results, the performance is significantly correlated with the fraction of outliers in the unlabeled data. If the

unlabeled data are all outliers (requiring known labels), DDAD boils down to a supervised method, which limits its wide application. All aforementioned reconstruction-based methods rely heavily on massive amounts of normal images, and in addition, require a high computational load due to the pixel-level comparison. Instead, our method is feature-based, which is free from the reconstruction of the whole image while achieves significant better performance with same amount of data.

Several works focus on devising self-supervised learning methods for anomaly detection [10]–[12]. The authors in [10] propose to train a multi-class model to discriminate between several geometric transformations applied on all the given images. The method in [12] additionally applies a set of pre-defined shifted transformations to images to create negative samples in the framework of contrastive learning. Meanwhile, a classification head is added to predict which shifting transformation is applied to the given image. However, classification-based methods require a sophisticated design of data transformations. The method in [11], inspired by [13], tries to learn the prototypical patterns of normal training samples and anomalous pattern via a memory bank and the anomaly score is calculated as a weighted combination of normal prototypical patterns. However, this method necessitates the availability of diverse augmented views of images as well as a limited number of labeled anomalous images for training. Our method replies solely on two random augmentations from the same augmentation distribution and does not require any labelled abnormal images.

Contributions In this work, a *feature-based* method for anomaly detection is proposed, which employs the structure of a Siamese network and consists of a pre-trained backbone, an adaptor layer, and a predictor layer. A schematic overview of the method is shown in Fig. 1. Our method 1) is entirely feasible to use various pre-trained backbones, 2) achieves state-of-the-art results compared to other reconstruction-based methods [4]–[6], [9], [14] when only using normal images for (semi-supervised) training, 3) is robust to different amounts of abnormal images among the normal images in the training data.

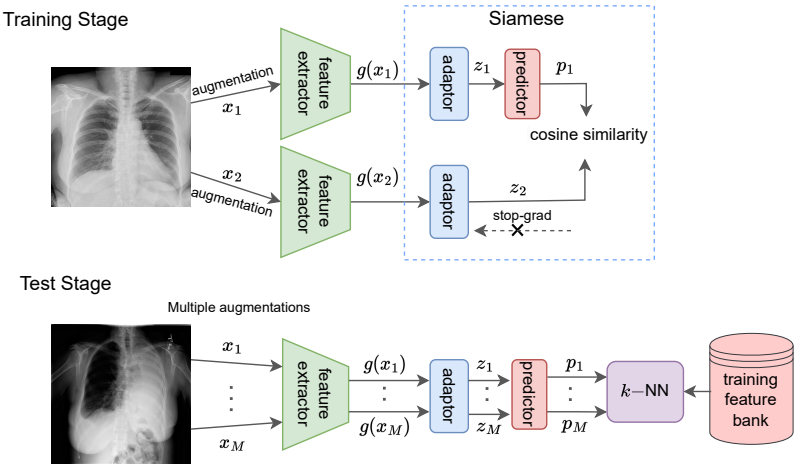

Figure 1: The proposed pipeline. In the training stage (top), two random augmentations are applied to the input images, and feature banks are learned using the Siamese architecture. At test time (bottom) the geometric mean of the k-NN distances between augmented test embeddings and training feature banks yields the anomaly score.

2 Method

In this work, we use a similarity measure of feature embeddings to compute the anomaly score for differentiating abnormal samples from normal samples. Firstly, it utilizes a powerful pre-trained backbone. This allows us to leverage existing well-established pre-trained models (e.g. on ImageNet-1k [15]). Secondly, we only need to learn light-weight feature adaptors on the domain of interest (i.e. chest X-ray images in our case) by employing the Siamese [16] architecture in a self-supervised way, where input images are the two augmented views of the same image. Importantly, our method does not necessitate a large batch size or the use of a pair of strong and weak augmentations. The training loss is the negative cosine similarity between feature embeddings of two augmented views of the input image. At test time, the geometric mean of k-nearest neighbor (k-NN) algorithm is applied to perform anomaly detection in the feature space. The proposed method is easy to implement and requires only minimal training (e.g. less sophisticated data transformations compared

to other self-supervised methods [10], [12]), but is nevertheless effective in detecting anomalies.

Pre-trained feature extractor. Deep models trained on ImageNet-1K are widely used as feature extractors in medical applications due to the scarcity of labelled data [17], [18]. Inspired by [19], [20], we use a network pre-trained in a self-supervised manner, since it has been empirically shown that self-supervised pre-training outperforms supervised pre-training, especially in semi-supervised settings [19].

Any self-supervised pre-trained model [16], [19], [21] can work as a feature extractor in our framework. In this work, we use two of the most popular pre-trained models, ResNet-50 trained by SimCLRv2 [19] and Barlow [21] on ImageNet-1k.

Training. The pre-trained feature extractor on ImageNet-1k might preserve the general representation of natural images. To obtain a domain-specific representation of the target data (CXRs), an adaptor layer is added on top of the pre-trained feature extractor to distill the knowledge of CXRs. We use the Siamese [16] architecture to learn the representation of normal samples in a self-supervised way. The training architecture is shown in Fig. 1. An input image x is perturbed by sampling two different augmentations from the same augmentation distribution, denoted as $\mathcal{T}$, yielding x_1 and x_2. In particular, we use random crops and horizontal flips as our pool of augmentations.

In the first stage, x_1 and x_2 are processed by the pre-trained feature extractor g. The resulting features are subsequently transformed by an adaptor network f (consisting of one fully connected layer) and one set of features is additionally processed by a predictor network h (a fully connected layer). Specifically, the two resulting feature vectors are given by $p_1 = h(f(g(x_1)))$ and $z_2 = f(g(x_2))$. The training loss is the negative cosine similarity $\mathcal{D}$:

$$\mathcal{D}(p, z) = -\left\langle \frac{p}{\|p\|}, \frac{z}{\|z\|} \right\rangle, \tag{D.1}$$

where z and p denote features from the adaptor layer and predictor layer of the two augmentation branches, respectively. As suggested in [16], a symmetrical loss $\mathcal{L}$ is used, and the stop-gradient technique is adopted to tackle the issue of requiring a momentum encoder, negative samples, or larger batches, which

are very common components in self-supervised training. The final training loss is given as follows,

$$\mathcal{L} = \tfrac{1}{2}\mathcal{D}\big(p_1, \text{stopgrad}(z_2)\big) + \tfrac{1}{2}\mathcal{D}\big(p_2, \text{stopgrad}(z_1)\big), \tag{D.2}$$

where the adaptor f receives only gradient information from the branch incorporating the predictor h. After the training process, the training feature bank $\mathcal{P}$ is constructed by applying multiple augmentations sourced from the augmentation distribution $\mathcal{T}$ to the training data. This is done with the aim of capturing a wider range of variations present in the training data.

Inference. At test time, an X-ray image is processed by the trained network and its feature representation from the predictor layer is used to compute an anomaly score. Specifically, we apply multiple augmentations drawn from the augmentation distribution $\mathcal{T}$ and obtain M augmented feature maps for the test image. To ensure the learned features are scale invariant, unit normalization is applied to the learned features before calculating the score. The score is calculated as the geometric mean of the k-NN using the Euclidean distance between the feature banks $\mathcal{P}$ and the feature extracted from the test image x^*. Specifically, let $(p_1^*, p_2^*, \cdots, p_M^*)$ be the features resulting from the multiple augmentations, then the anomaly score S is defined as

$$S(x^*) = \sqrt[M]{\|p_1^* - \mathcal{P}^{[k]}\| \cdot \|p_2^* - \mathcal{P}^{[k]}\| \cdots \|p_M^* - \mathcal{P}^{[k]}\|}, \tag{D.3}$$

where $\|p_i^* - \mathcal{P}^{[k]}\|$ is the Euclidean distance to the k-th NN in feature bank $\mathcal{P}$. The k-NN computation is performed independently for each augmentation.

3 Experiments

Datasets. We evaluate our method on the RSNA Pneumonia Detection Challenge dataset [1] and the VinBigData Chest X-ray Abnormalities Detection dataset [2], and we use the exact the same split between training and testing as in [9] to enable a fair comparison. The performance is evaluated by the area under the receiver operating characteristic curve (AUROC) and average

[1] https://www.kaggle.com/c/rsna-pneumonia-detection-challenge
[2] https://www.kaggle.com/c/vinbigdata-chest-xray-abnormalities-detection

precision (AP). The **RSNA dataset** includes 8,851 normal and 6,012 abnormal images. 1,000 normal and 1,000 abnormal images (i.e. lung opacity) are combined to create the RSNA test set. The **VinBigData dataset** is a much more challenging dataset, which consists of 10,601 normal and 4,394 abnormal images that cover 14 types of thoracic abnormalities (e.g. calcification and pleural thickening). The VinBigData test data includes 1,000 normal images as well as 1,000 abnormal images.

Data Augmentation. All input images are resized to 224×224×3. Since the original feature extractor trained by simCLRv2 [19] and Barlow [21] uses RGB images, we create 3-channel images by duplicating the 1-channel grayscale images. In our approach, the success of training does not rely on the use of strong and weak augmentations. Instead, we utilize two augmentations from the same augmentation distribution $\mathcal{T}$, i.e. random crops and random horizontal flips. During inference, the training feature bank $\mathcal{P}$ is M times size of the original training data, where M is the number of augmentations performed on each training image.

Implementation details. The architecture of the pre-trained feature extractor is ResNet50 trained by SimCLRv2 [19] and Barlow [21] on ImageNet-1k in our experiment. The adaptor of the proposed models consists of one fully connected layer with the size of $(2048 \rightarrow 1024)$, and the predictor consists of one fully connected layer with the size of $(1024 \rightarrow 1024)$. Our model is trained for 100 epochs using the Adam optimizer with a learning rate 10^{-5} for simCLRv2 [19], and $2 \cdot 10^{-7}$ for Barlow [21]. The number of augmentations applied is $\boldsymbol{M = 5}$ and $\boldsymbol{k = 1}$. All experiments were run on a single NVIDIA GeForce RTX 2080Ti, CUDA 11.2, using PyTorch 1.9.0+cu111 **pyt**. The inference time per image is approximately 50ms for both datasets.

3.1 Experimental results

Comparison with SOTA methods. We first compare our method trained on only normal images with a line of reconstruction-based methods including AE [14], MemAE [5], f-AnoGAN [6], AE-U [4], and DDAD-AE-U [9]. DDAD-AE-U refers to AE-U combined with the DDAD method. Due to the special setting of DDAD-AE-U [9], all methods are trained with partial training data,

Methods	RSNA		VinBigData	
	AUROC ↑	AP ↑	AUROC ↑	AP ↑
AE [14]	0.669	-	0.559	-
MemAE [5]	0.680	-	0.558	-
f-AnoGAN [6]	0.798	-	0.763	-
AE-U [4]	0.867	-	0.738	-
DDAD-AE-U [9]	0.873	-	0.743	-
SimCLRv2* (Ours)	<u>0.882</u>	<u>0.863</u>	**0.846**	**0.824**
Barlow* (Ours)	**0.905**	**0.908**	<u>0.809</u>	<u>0.802</u>

Table 1: *Comparison with SOTA methods* (using $DR = 0.49$ for RSNA and $DR = 0.5$ for VinBigData). Values are AUROC and AP, where boldface indicates the best, underline indicates the second best. $*$ represents adding the proposed adaptor layer and predictor layer.

following the same dataset splits as in [9], i.e. implemented using their provided data list. Specifically, 3,851 normal images and 4,000 normal images are used as training data [3] for the RSNA dataset and the VinBigData dataset, respectively. The results in Table. 1 show that our method consistently outperforms the other semi-supervised methods [4] on both datasets. In particular, by AUROC our method surpasses f-AnoGAN, which is SOTA for the more challenging VinBigData dataset, with a large margin of 8.3%. When using inter-discrepancy as a score, DDAD-AE-U is essentially a supervised method and these results are therefore not included in our comparison.

Different training data amount. We explored the influence of different amounts of clean training data on our anomaly detection method's performance by training the network with various training data ratios (DR) relative to the total amount of training data (n=X). Specifically, we selected DR values of $10\%, 30\%, 50\%, 70\%$, and 90%. The results for the two datasets are shown in Fig. 2. For the RSNA dataset, even with only 30% of the training data, our method could obtain much better results (AUROC = 0.902) than DDAD-AE-U [9] with 49% of the training data (AUROC = 0.873). For the VinBigData

[3] 3951 images in RSNA corresponds to $DR = 0.49$, 4000 images in VinBigData corresponds to $DR = 0.5$.

[4] We use the results reported in [9].

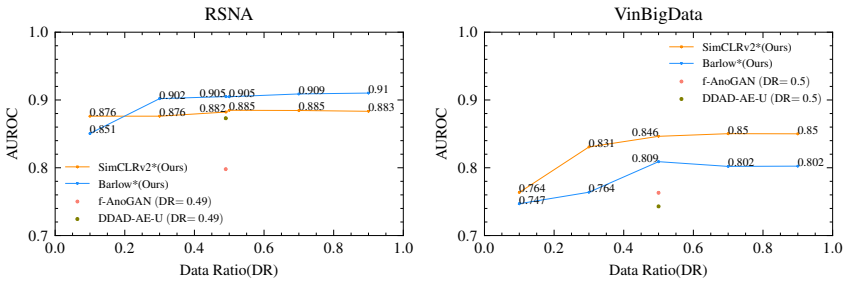

Figure 2: The performance of our method varies with the amount of training data.

dataset, the figure indicates that our method could achieve significantly better results even with 30% of the training data (AUROC = 0.831) compared with SOTA f-AnoGAN [6] with 50% of the training data (AUROC = 0.763). Notably, as the number of training images increases, the performance on both datasets with different backbones appears to reach a saturation point. One possible explanation for this observation is that the augmentations applied to the training/test data already encompass a wide range of variations, leaving little room for further improvement. Additional information regarding the impact of varying the number of augmentations can be found in the supplementary material.

Different anomaly amount. A more realistic setting is that the training data includes both normal and some amount of anomalous data, resembling unsupervised learning conditions. Hence, it is crucial that the proposed method is robust to different amounts of abnormal images in the training data, and therefore, we examine the influence of varying anomaly ratios (AR). The results can be found in Fig. 3. The results of the baseline models including f-AnoGAN and DDAD-AE-U corresponds to $AR = 0$, i.e. only known normal images in the training data. For the RSNA dataset, our method Barlow* with a 10% anomaly ratio achieves higher AUROC value than DDAD-AE-U [9] without any anomalies in the training data. Our method, regardless of the anomaly ratio, consistently obtain better results on the VinBigData compared to the DDAD-AE-U and f-AnoGAN methods trained without anomalies.

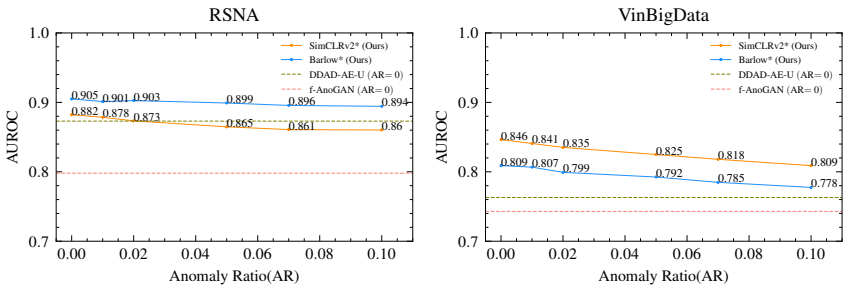

Figure 3: The performance of our method varies with the amount of anomalies in the training data. (using $DR = 0.49$ for RSNA and $DR = 0.5$ for VinBigData.)

Importance of model components. We evaluate the relevance of the following three components of our method: using the pre-trained feature extractor (i.e. ResNet50 trained by simCLRv2), training with Siamese style, and the necessity of feature normalization in the feature banks $\mathcal{P}$. The impact of each component is shown in Table. 3. No pre-trained means that the pre-trained backbone is replaced with a trainable encoder of the same size as the baseline AE [9], and no Siamese corresponds to directly using features extracted from the pre-trained backbone without fine-tuning on the target dataset, i.e. removing the adaptor and predictor. Finally, the impact of adding scale invariance by feature normalization to the stored features in $\mathcal{P}$ is investigated.

It is noteworthy that training the proposed layers significantly improve the performance on the VinBig dataset compared to the RSNA dataset. This could be attributed to the fact that the anomalies present in the RSNA dataset are limited to lung opacities, and the pre-trained models already possess sufficient capability to differentiate them from healthy images. Conversely, the diverse nature of anomalies in the VinBigData, making them more challenging to distinguish. Similar observations were made when using Barlow as the backbone. For additional results using Barlow [21] as the backbone can be found in supplementary material.

Backbone	Pre-trained	Siamese	Normalization	RSNA		VinBigData	
				AUROC ↑	AP ↑	AUROC ↑	AP ↑
simCLRv2	✗	✓	✓	0.689	0.651	0.678	0.666
	✓	✗	✓	0.754	0.688	0.623	0.619
	✓	✓	✗	0.798	0.776	0.813	0.800
	✓	✓	✓	**0.885**	**0.878**	**0.846**	**0.824**

Table 2: *The importance of different model components* (using $DR = 0.5$ and $AR = 0$) for the two datasets. Values are AUROC and AP.

4 Conclusion and future work

In this work, we propose a feature-based method for anomaly detection, comprising of a pre-trained backbone extended with an adaptor and a predictor layer. Our approach is simple, easy to train, and exhibits improved performance compared to reconstruction-based methods in a semi-supervised setting. The method is versatile by allowing the use of various backbones, and the suggested adaptor and predictor layers are shown to enhance anomaly detection performance. Additionally, the proposed method is able to cope with varying levels of outliers (abnormal images) in the training data, making it suitable for realistic unsupervised learning conditions. We train and test our model to detect medical anomalies, but the method can with ease be applied to more general and diverse outlier detection tasks as well, e.g. detecting non-pathological anomalies such as image artifacts or artificial implants. Furthermore, the current applied augmentation are the most basic type of augmentations, it is also interesting to investigate various augmentations tailored to specific target anomalies, aiming to enhance the performance.

5 Supplementary Material

I Effective number of augmentations

The impact of different number of augmentations applied to the test image is investigated. Specifically, a fixed training feature bank $\mathcal{P}$ is obtained by applying 5 augmentations sourced from the augmentation distribution $\mathcal{T}$ to the training data. Different number of augmentations $M = \{2, 5, 10\}$ are applied to each test image and $k = 1$, the corresponding results of RSNA dataset and VinBigData can be found in Fig 4. Empirically, larger values of M yield improved performance in terms of both AUROC and AP.

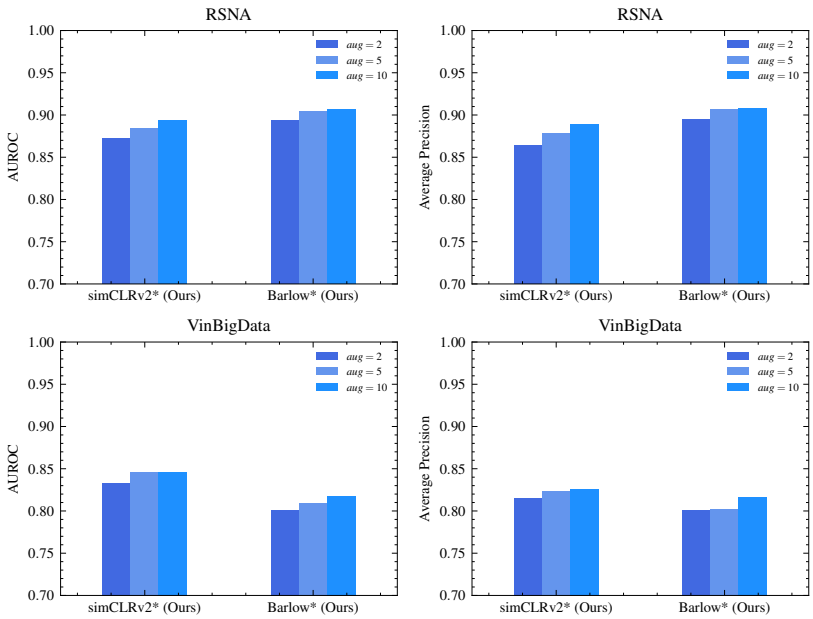

Figure 4: *Effective value of M.* The performance of our method varies with different number of augmentations applied to test images (using $DR = 0.5, AR = 0$).

II Importance of components

Backbone	Pre-trained	Siamese	Normalization	RSNA		VinBigData	
				AUROC ↑	AP ↑	AUROC ↑	AP ↑
Barlow	✗	✓	✓	0.689	0.651	0.678	0.666
	✓	✗	✓	0.903	0.898	0.785	0.773
	✓	✓	✗	0.708	0.668	0.603	0.563
	✓	✓	✓	**0.905**	**0.907**	**0.809**	**0.802**

Table 3: *The importance of different model components* (using $DR = 0.5$ and $AR = 0$) for the two datasets. Values are AUROC and AP.

III Average precision of different amount of training data

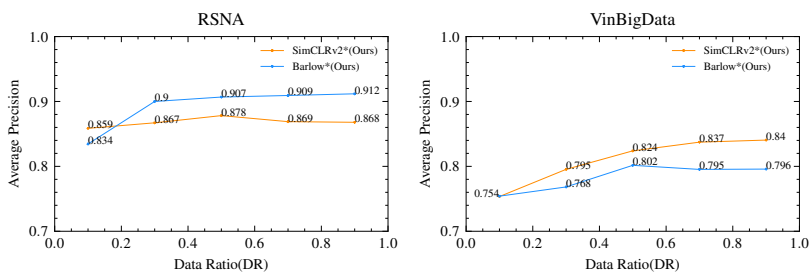

Figure 5: The performance of our method varies with the amount of training data.

IV Average precision of different amount of anomaly data

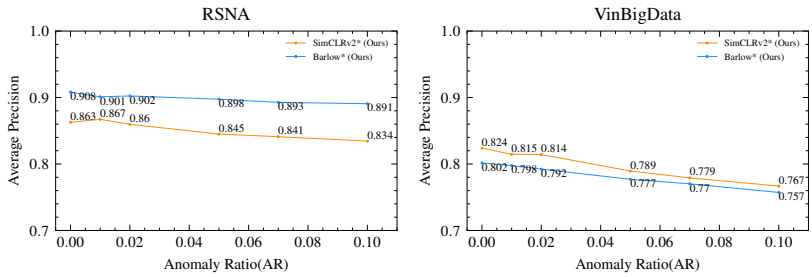

Figure 6: The performance of our method varies with the amount of anomalies in the training data.(using DR = 0.5)

References

[1] E. Çallı, E. Sogancioglu, B. van Ginneken, and K. G. van, "Deep learning for chest x-ray analysis: A survey," *Medical Image Analysis*, 2021.

[2] T. Fernando, H. Gammulle, S. Denman, S. Sridharan, and C. Fookes, "Deep learning for medical anomaly detection – a survey," arXiv, 2020.

[3] T. Cao, C. Huang, D. Y. Hui, and J. P. Cohen, "A benchmark of medical out of distribution detection," *Journal of Machine Learning for Biomedical Imaging*, 2020.

[4] Y. Mao, F.-F. Xue, R. Wang, J. Zhang, W.-S. Zheng, and H. Liu, "Abnormality detection in chest x-ray images using uncertainty prediction autoencoders," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2020.

[5] D. Gong, L. Liu, V. Le, *et al.*, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[6] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurth, "F-anogan: Fast unsupervised anomaly detection with generative adversarial networks," *Medical Image Analysis*, 2019.

[7] J. Wolleb, F. Bieder, R. Sandkühler, and P. C. Cattin, "Diffusion models for medical anomaly detection," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2022.

[8] J. Wyatt, A. Leach, S. M. Schmon, and C. G. Willcocks, "Anoddpm: Anomaly detection with denoising diffusion probabilistic models using simplex noise," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2022.

[9] Y. Cai, H. Chen, X. Yang, Y. Zhou, and K.-T. Cheng, "Dual-distribution discrepancy for anomaly detection in chest x-rays," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2022.

[10] I. Golan and R. El-Yaniv, "Deep anomaly detection using geometric transformations," in *Advances in Neural Information Processing Systems*, 2018.

[11] B. Bozorgtabar, D. Mahapatra, G. Vray, and J.-P. Thiran, "Salad: Self-supervised aggregation learning for anomaly detection on x-rays," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2020.

[12] J. Tack, S. Mo, J. Jeong, and J. Shin, "Csi: Novelty detection via contrastive learning on distributionally shifted instances," in *Advances in Neural Information Processing Systems*, 2020.

[13] C. Zhuang, A. L. Zhai, and D. Yamins, "Local aggregation for unsupervised learning of visual embeddings," 2019.

[14] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "Ganomaly: Semi-supervised anomaly detection via adversarial training," in *Asian Conference on Computer Vision (ACCV)*, 2018.

[15] O. Russakovsky, J. Deng, H. Su, *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, 2015.

[16] X. Chen and K. He, "Exploring simple siamese representation learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[17] H. Xie, H. Shan, W. Cong, *et al.*, "Dual network architecture for few-view CT – trained on imagenet data and transferred for medical imaging," in *International Society for Optics and Photonics*, 2019.

[18] S. M. McKinney, M. Sieniek, V. Godbole, *et al.*, "International evaluation of an AI system for breast cancer screening," *Nature*, 2020.

[19] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton, "Big self-supervised models are strong semi-supervised learners," in *Advances in Neural Information Processing Systems*, 2020.

[20] S. Azizi, B. Mustafa, F. Ryan, *et al.*, "Big self-supervised models advance medical image classification," 2021.

[21] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in *International Conference on Machine Learning (ICML)*, 2021.

PAPER E

TAG: Text Prompt Augmentation for Zero-Shot Out-of-Distribution Detection

Xixi Liu, Christopher Zach

Published in European Conference on Computer Vision (ECCV)
pp. 364-380, 2024
©DOI: 10.1007/978-3-031-73464-9-22

The layout has been revised.

Abstract

Out-of-distribution (OOD) detection has been extensively studied for the reliable deployment of deep-learning models. Despite great progress in this research direction, most works focus on discriminative classifiers and perform OOD detection based on single-modal representations that consist of either visual or textual features. Moreover, they rely on training with in-distribution (ID) data. The emergence of vision-language models allows to perform zero-shot OOD detection by leveraging multi-modal feature embeddings and therefore only rely on labels defining ID data. Several approaches have been devised but these either need a given OOD label set, which might deviate from real OOD data, or fine-tune CLIP, which potentially has to be done for different ID datasets. In this paper, we first adapt various OOD scores developed for discriminative classifiers to CLIP. Further, we propose an enhanced method named *TAG* based on Text prompt AuGmentation to amplify the separation between ID and OOD data, which is simple but effective, and can be applied on various score functions. Its performance is demonstrated on CIFAR-100 and large-scale ImageNet-1k OOD detection benchmarks. It consistently improves AUROC and FPR95 on CIFAR-100 across four commonly used architectures over four baseline OOD scores. The average AUROC and FPR95 improvements are 6.35% and 10.67%, respectively. The results for ImageNet-1k follow a similar, but less pronounced pattern. The code is available at: `https://github.com/XixiLiu95/TAG`.

1 Introduction

To guarantee the safe deployment of deep learning models in the "wild," particularly for high-stake applications such as autonomous driving [1] and intelligent health care [2], it is unarguably critical for the models to learn what they do not know [3]. For instance, models should be able to flag inputs highly unlikely according to the training distribution and avoid unreliable

predictions for such data. Specifically, models are expected to identify samples that exhibit covariate shift (change in the input distribution) or semantic shift (change in the label distribution) depending on the use case [4].

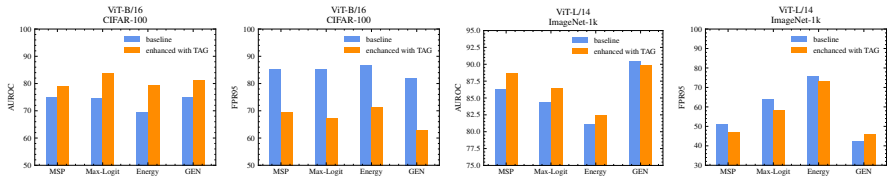

Figure 1: *Effectiveness of TAG applied with 4 Baseline Scores on CIFAR100 (left 2 columns) and ImageNet-1k (right 2 columns).* Reported are AUROC values (%) and FPR95 (%). The averages are computed across 5 OOD datasets (CIFAR-100) and 4 OOD datasets (ImageNet-1k), respectively.

In this work, we focus on the case of identifying semantic shift. A plethora of the research in this setting uses standard discriminative classifiers [5]–[16], where the label information is simply encapsulated as a one-hot vector. Therefore, the above-mentioned methods mostly rely on visual features extracted from the pre-trained models. Moreover, their OOD detection performance is highly correlated with the accuracy of the classifiers [10], [11]. The emergence of vision-language models (VLMs e.g. CLIP [17]) that learn the joint representation of image and text offers a great opportunity to exploit it for OOD detection, particularly, for the semantic shift. Specifically, the text prompt for each class, processed by the text encoder, can be viewed as a class prototype (in feature space). Unlike scenarios involving discriminative classifiers, where the models must undergo training on the ID dataset, CLIP-based methods only require the set of labels comprising the ID dataset. A number of works have observed that CLIP [17] can be used as a powerful zero-shot OOD detector [18]–[22]. Following the definition in [19], zero-shot OOD detection means that only the names of the ID dataset can be utilized, and it does not access the training data of ID dataset. However, some of them either need to create the OOD label set manually [18] or generate the OOD label set automatically [19], which might diminish the OOD performance if the designed OOD label is not representative. MCM [20] is free from the pre-defined OOD labels but less effective in some cases. GL-MCM [21] enhances the performance of MCM by exploiting the local features, which to some extent restricts its de-

ployment scenarios. [22] designs a pipeline for OOD detection by utilizing the external knowledge from large language models (LLMs) to generate descriptors for the ID dataset. In this work, we extend the score functions based on a single-modal regime (i.e., discriminative classifiers) to a multi-modal regime (i.e., CLIP [17]) to perform zero-shot OOD detection. Furthermore, an enhanced method based on text prompt augmentation is proposed to further improve the performance. Figure. 1 highlights its performance compared to other baseline score functions.

Contribution We present a simple but effective method, *TAG* (for Text prompt AuGmentation), to enhance the performance of zero-shot OOD detection equipped with various score functions including MSP [6], MaxLogit [7], Energy [8], and GEN [11].

1. TAG only uses label information of the training data and is completely outlier-free (in terms of both OOD data and label information). It also does not require external knowledge from LLMs, sophisticated prompt ensembling or additional training, meaning it can be deployed in a wider range of scenarios.

2. It consistently achieves significantly better results under various score functions on CIFAR-100, and the improvement remains on ImageNet-1k across 4 architectures and 3 baseline OOD methods (Fig. 1 and Section 4).

2 Related Work

Vision-based OOD detection

Performing OOD detection in terms of semantic shift on discriminative classifiers has been a long-standing research field [6]–[9], [11], [13]–[16], [23]–[32], and can be roughly categorized based on whether the outliers are exposed during training. Firstly, the methods that do not require outlier exposure (OE) can be grouped into (i) deriving new score functions based on either logit information such as Energy [8] and MaxLogit [7], or predictive distribution such as MSP [6] and GEN [11]; additionally, GradNorm [9] utilizes the information from both features extracted from the penultimate layer and predictive distributions. (ii) utilizing the training feature statistics such as [5],

[10] or the learned weight of the last fully connected layer [16] to devise OOD score. It is intuitive that using the information from the training data could further boost the performance of OOD detection. However, this is infeasible in the case when the training data is confidential or otherwise unavailable. (iii) enhancing the OOD performance by either obtaining distinct features to distinguish ID and OOD data such as ODIN [33], Generalized ODIN [26], ReAct [13], RankFeat [12], ASH [30], and SCALE [31], or augmenting softmax-based confidence scores with feature-agnostic information such as SIRC [25]. Those enhanced methods are compatible with several score functions including MSP [6], Energy [8], and GEN [11]. Additionally, unlike the training of a standard classifier using cross-entropy loss, [24] and CIDER [27] devise contrastive learning-based methods for OOD detection.

The methods required to access OOD data typically involve devising a new training loss with OE explicitly [14], [28] or implicitly [15], [29], [32], [34]. Specifically, [14] firstly propose to jointly optimize a classification loss and a regularization term that forces the predictive distribution of the OOD sample to be uniform. [28] proposes to perform outlier mining firstly by sampling a posterior distribution and then applying energy regularization [8] afterward. Additionally, [34] argues that the selected OOD data for training might deviate from the real OOD data and the performance of OE might degrade on the unseen OOD data. Therefore, a min-max learning scheme is formulated to search for the OOD samples that are most intriguing to the model and learn from such OOD data. However, heavier computation is required compared to other OE methods. [29] does not rely on any OOD data but instead obtains the OOD feature embeddings by sampling the low density of the training feature space. While [15] utilizes the learned text embeddings of the training data and draws samples from the low-density regime to obtain OOD text embeddings. Furthermore, the sampled OOD text embeddings are processed with Stable Diffusion [35] to generate synthetic OOD samples. Finally, energy regularization [8] is applied to enable the training for OOD detection. Nevertheless, implementing this method requires generating OOD data, in particular, for each ID dataset, thereby its applicability is restricted in various deployment scenarios.

Vision-language based OOD detection

CLIP [17], as the most popular and publicly available VLM is getting recognition for the task of OOD detection [18]–[20], [36]. [18] is the first work to explore the capability of CLIP [17] for zero-shot OOD detection. Specifically, two non-overlapped sets of label space including the class names of the ID dataset $\mathcal{Y}_{\text{ID}}$, and class names manually designed $\mathcal{Y}_{\text{OOD}}$ are created. During inference, an image embedding $\mathcal{I}(\text{x})$ is obtained for each image x, and applying Softmax to the logits s (i.e., the cosine similarity between the image embedding $\mathcal{I}(\text{x})$ and all text embeddings), the predictive distribution is obtained and denoted by $\text{p} = \text{Softmax}(\text{s})$. Note p can be split to $p(\text{in}|\,\text{x}) = \sum_{i\in\mathcal{Y}_{\text{ID}}} \text{p}_i$ and $p(\text{out}|\,\text{x}) = \sum_{i\in\mathcal{Y}_{\text{OOD}}} \text{p}_i$, and $p(\text{in}|\,\text{x}) + p(\text{out}|\,\text{x}) = 1$. Finally, the OOD score is designed as $p(\text{in}|\,\text{x}) = \sum_{i\in\mathcal{Y}_{\text{ID}}} \text{p}_i$. To resolve the inconvenience of manually designed OOD labels arising from [18], ZOC [19] instead trains a text description generator to obtain $\mathcal{Y}_{\text{OOD}}$ automatically. First, a text-decoder denoted by $\text{Decoder}_{\text{text}}$ is trained on a large captioning data (i.e., a set of paired images and texts.). Afterward, the pre-trained $\text{Decoder}_{\text{text}}$ is used to generate an image description for each test image and then the top k words from the vocabulary with the highest probabilities are selected as $\mathcal{Y}_{\text{OOD}}$. The final label space is $\mathcal{Y}_{\text{ID}} \cup \mathcal{Y}_{\text{OOD}}$. The way to obtain the predictive distribution is the same as [18], but the final OOD score is defined as $1 - \sum_{i\in\mathcal{Y}_{\text{ID}}} \text{p}_i$. Although [18], [19] demonstrated superior performance on OOD detection, they both rely on pre-defined OOD label sets, which unavoidably impedes their performance as the defined OOD labels might deviate from the real OOD label. Unsatisfactorily, the OOD label set potentially has to be designed for every ID dataset. Instead, CLIPN [36] fine-tunes the CLIP [17] by introducing an additional text encoder on par with negative (learnable) prompts. The training loss incorporates two key components: image-text binary-opposite loss, which aims to align the image embedding with its unrelated negative text embedding, and the text semantic-opposite loss, designed to maximize the l_2 distance between two text embeddings with opposing meanings. The final OOD score is calculated either through the competing-to-win (CTW) algorithm or through the agreeing-to-differ (ATD) algorithm. However, the fine-tuning of CLIP [17] inevitably has to be done for each ID dataset. MCM [20] instead neither depends on the design of the OOD label nor requires additional fine-tuning. It directly uses the text embeddings processed from the prompts `this is a photo of a` $\langle y_k \rangle$ as the concept prototypes to perform OOD detection. Our method

TAG does not require both pre-defined OOD labels and pre-training. Moreover, it can be applied to MCM [20], potentially enhancing the performance of OOD detection.

Prompt engineering with external knowledge

To improve the performance of zero-shot visual classification using VLMs, DCLIP [37] extends the default prompt for each class with its corresponding descriptions generated by LLMs (e.g., GPT-3). Instead, WaffleCLIP [38] empirically shows that replacing the generated GPT-3 descriptions with random word or character sequences leads to competitive performance. [22] explores to design a multi-modal OOD framework by utilizing the external knowledge from LLMs. However, additional calibration methods are required to maintain the quality of generated descriptors because of the hallucination of LLMs [39]. Different from [37], [38], our method is devised for the task of OOD detection and solely rely on the default prompt without any external knowledge. Moreover, our method can be integrated with DCLIP [37] and WaffleCLIP [38], potentially enhancing the performance of OOD detection.

3 Text Prompt Augmentation

CLIP [17] is a vision-language model and consists of a text encoder $\mathcal{T}$ and an image encoder $\mathcal{I}$. Hundreds of millions of paired images and texts equipped with InfoNCE [40] loss are used for its training. To perform OOD detection using CLIP [17] for a given ID dataset denoted by $\mathcal{D}_{\text{in}}$ with label space denoted by $\mathcal{Y}_{\text{in}} = \{y_1, y_2, \cdots, y_K\}$, the default text prototype t_k for the class k can be constructed as `a photo of` $\langle y_k \rangle$. During inference, a test image x is firstly processed by image encoder $\mathcal{I}$, we can re-interpret the cosine similarity s_k between extracted feature $\mathcal{I}(\mathrm{x})$ and all text prototypes $\mathcal{T}(t_k)$ as the logit, which is further normalized by Softmax, the probability that image x belong to class k can be calculated as

$$p_k(\mathrm{x} \,|\mathcal{Y}_{\text{in}}, \mathcal{I}, \mathcal{T}) = \frac{\exp(s_k/\tau)}{\sum_{j=1}^{K} \exp(s_j/\tau)}, \tag{E.1}$$

where $s_k = \frac{\mathcal{I}(\mathrm{x}) \cdot \mathcal{T}(t_k)}{\|\mathcal{I}(\mathrm{x})\| \cdot \|\mathcal{T}(t_k)\|}$, and τ is a temperature parameter. By this interpretation, various score functions such as MSP [6], MaxLogit [7], Energy [8],

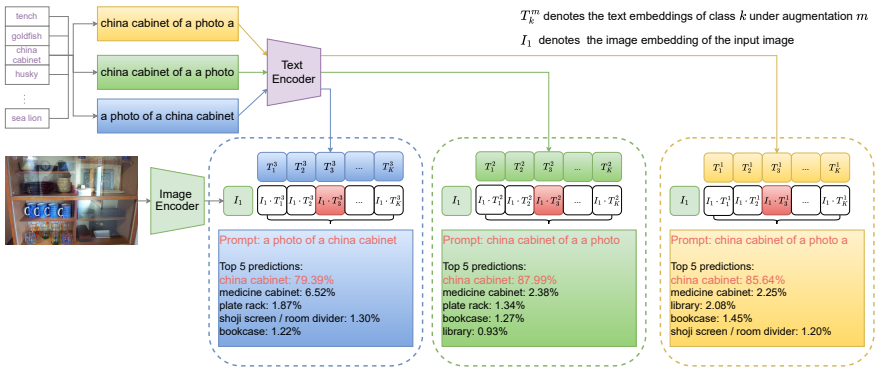

Figure 2: *Probabilities of Top-5 Predictions Using Different Sequences of the Default Text Prompt.* Class names are taken from ImageNet-1k and ViT-B/16 is used as backbone. The shuffled prompts of the non-target class are omitted for clean visualization.

and GEN [11] developed for the discriminative classifier can be applied to CLIP [17] to perform OOD detection. The most significant benefit of using CLIP [17] is that there is no need to access the training data of the ID dataset since the set of semantic labels for the ID dataset is the sole requirement.

Sequence of the prompt

The default text prompt for CLIP [17] includes but is not limited to `a photo of a` $\langle y_k \rangle$. We observe that it is not necessary to use the right order of the text prompt. Instead, with a grammatically incorrect sequence $\langle y_k \rangle$ `of a a photo`, CLIP [17] may still yield a correct classification, sometimes even with a higher probability for the target class. An example is illustrated in Figure 2. Here the default prompt is randomly shuffled and a classification task on ImageNet-1k [42] is performed based on the shuffled prompt. One can see that the image of the china cabinet is correctly classified with higher probability using the incorrect order of text prompt.

Effect of text prompt augmentation

It is empirically observed that the cosine similarity is non-uniform for the ID dataset, which is also noticed by MCM [20]. Moreover, we also observe that

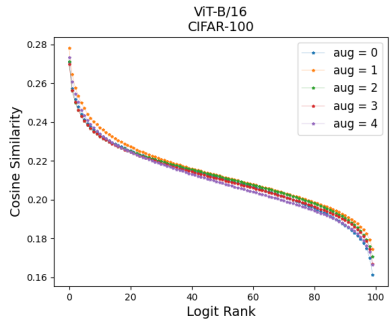

Figure 3: *Average Sorted Cosine Similarity.* The dataset is CIFAR-100 [41], and $M = 5$ different text prompt augmentations are applied.

this phenomenon consistently occurs when different text augmentations are applied. The average cosine similarity of CIFAR100 applied with 5 different random text augmentations is shown in Figure 3.

TAG

Motivated by the aforementioned phenomenon, an enhanced method is proposed to improve the performance of OOD detection under various score functions. Specifically, M augmented text prompts for each class k can be obtained by randomly shuffling the default prompt that CLIP [17][1] uses. The PyTorch-like code for generating M different augmented tokens (i.e. tokenized text prompt) is presented in Algorithm. 1. Each augmented set is denoted by $t^m = \{t_1^m, t_2^m, \cdots, t_K^m\}$, where t_k^m denotes the augmented text prompt for class k under augmentation m. After obtaining M sets of text prompts, the probability that the test sample x belonging to class k with the text prompt augmentation t_k^m is calculated as

$$p_k^m(\mathrm{x} \,|\mathcal{Y}_{\text{in}}, \mathcal{I}, \mathcal{T}) = \frac{\exp(s_k^m/\tau)}{\sum_{j=1}^{K} \exp(s_j^m/\tau)}, \tag{E.2}$$

where $s_k^m = \frac{\mathcal{I}(\mathrm{x}) \cdot \mathcal{T}(t_k^m)}{\|\mathcal{I}(\mathrm{x})\| \cdot \|\mathcal{T}(t_k^m)\|}$ is the logit of class k with text-prompt m, and τ is the temperature hyper-parameter. Assuming MSP [6] is used as the OOD

[1] `a photo of a` $\langle y_k \rangle$

Algorithm 1: Generation of augmented tokens

```
# M: number of augmentations applied to the text prompt
# dataset: the ID dataset
def ShufflePrompt(words, c):
    random.shuffle(words) #  Shuffle the words randomly
    shuffled = ' '.join(words) # Reconstruct the shuffled prompt
    shuffled = shuffled.replace("classname", c)
    return shuffled
# Ensure that multiple-word class names are not split after shuffling
prompt = "a photo of a classname"
words = prompt.split() # Tokenize the prompt into words
MShuffledToken= [ ]
for m in range(M):
    TokenShuffled = [ ]
    for c in dataset.classes:
        text = ShufflePrompt(words, c)
        TokenShuffled.append(clip.tokenize(text))
    AllToken = torch.cat(TokenShuffled)
    MShuffledToken.append(AllToken)
```

score to perform OOD detection, meaning

$$S^m(\mathrm{x}) = \max_k p_k^m, \tag{E.3}$$

the final score function for OOD detection is

$$S(\mathrm{x}) = \tfrac{1}{M} \sum_{m=1}^{M} S^m(\mathrm{x}). \tag{E.4}$$

The alternative scoring methods including MaxLogit [7], Energy [8], and GEN [11] can also be utilized by substituting the Eq. E.3 with the respective score functions.

Logits vs. probabilities

In [20] it is argued that using the maximum probability (MSP/MCM) instead of the maximum logit (MaxLogit) is beneficial in terms of the FPR (Theorem 1 in [20]). In particular, for a sufficiently large choice of τ, MSP/MCM always yields a lower FPR than MaxLogit (under a certain assumption on the values of the non-maximal logits). In the supplementary material we improve on

their result by replacing the specific assumption on the logits (Assumption A.1 in [20]) with a simple assumption that the logits are bounded from below. This assumption is clearly satisfied for logits obtained as the cosine similarity between embedding vectors as they are constrained to the range $[-1, 1]$ by construction. We also want to point out that these theoretical results should be understood with some caution as by increasing τ only the FPR is controlled but not the TPR. This implies that very large values for τ will eventually be detrimental for the TPR, and a universal advantage of MSP/MCM over MaxLogit is not established.

4 Experiments

All experiments are conducted on two OOD benchmarks including CIFAR-100 [41] and ImageNet-1k [42]. We closely follow the evaluation protocol conducted in [15], [27] with the CIFAR-100 as the ID dataset. For ImageNet-1k [42], we follow the evaluation done by ViM [10] and GEN [11]. All pre-trained checkpoints of CLIP models including ViT-based and ResNet-based are provided by OpenAI[2].

Models

CLIP [17] is used to demonstrate the effectiveness of our method. We use 5 models released by CLIP [17], which can be grouped into 1) ViT-based models including ViT-B/16, ViT-B/32, and ViT-L/14, in which the vision transformer (ViT) is used as the image encoder. 2) ResNet-based models including ResNet-50 and ResNet-101, in which the ResNet is taken as the image encoder. The text encoders are either a Continuous Bag of Words (COBW) model or a text transformer.

Datasets

We perform OOD detection on a small-scale dataset with CIFAR-100 [41] as the ID dataset and a more realistic large-scale dataset with ImageNet-1k as the ID dataset. While CIFAR-100 has fewer classes compared to ImageNet-1k, the objects in the images are commonly centered and apparent. However, the objects in ImageNet-1k are sometimes rather small and sometimes partially

[2] `https://github.com/openai/CLIP`

occluded. For CIFAR-100 as ID dataset, the corresponding five OOD datasets are SVHN [43], iSUN [44], Places365 [45], Textures [46], and LSUN [47]. For the ImageNet-1k [42] as ID dataset, four commonly-used challenging OOD datasets are employed including ImageNet-O [48], Open-Image-O [49], Textures [46], and iNaturalist [50].

Score functions

Several commonly-used score functions derived for discriminative classifiers including MSP [6], MaxLogit [7], Energy [8], and GEN [11] are selected as the baseline methods. As suggested by GEN [11], we use top 100 classes and set $\gamma = 0.1$. Moreover, the score function MCM [20] (i.e. MSP with $\tau = 1$) designed for multi-modal models is also selected as one of the baselines.

Evaluation metrics

The area under the receiver operating characteristic curve (AUROC) and FPR95 — the false positive rate when the true positive rate is 95%- are commonly utilized for the evaluation of OOD detection. Higher values of AUROC indicate better performance and lower values of FPR95 are better. The reported units for both metrics in all tables are percentages.

4.1 OOD Detection Experimental Results

In this section, the results of OOD detection using four score functions devised for discriminative classifiers but adapted to CLIP [17] are presented first. Additionally, the score function MCM [20] designed for CLIP is also presented. Furthermore, the results of OOD detection enhanced with TAG denoted with $*$ are reported for each baseline score function. The experiments are running on NVIDIA GeForce RTX 2080Ti, CUDA 11.2 + PyTorch 2.1.0.

Results on CLIP-ViT-L/14 and CLIP-ResNet-101

Two OOD benchmarks are selected to perform OOD detection. The results of CIFAR-100 are shown in Table. 1. First, the first block in Table. 1 indicates that our method (TAG) consistently and significantly improves the performance of OOD detection under five different scores in terms of FPR95. Moreover, the performance gain is also present for ResNet-101 by looking at

OOD method	SVHN		iSUN		Places365		Textures		LSUN		Average	
	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓
ViT-L/14												
MSP [6]	90.54	51.91	84.23	75.26	65.52	95.99	72.11	91.37	83.03	75.71	79.09	78.05
MSP*	92.91	34.31	89.77	**50.94**	69.47	90.08	77.19	80.6	86.27	**64.34**	83.12 (↑4.03)	64.05 (↓14.0)
MaxLogit [7]	88.62	69.28	82.97	80.32	92.39	33.97	91.01	38.09	62.72	93.97	83.46	70.85
MaxLogit*	88.17	69.34	84.06	77.8	92.88	32.29	**91.16**	**37.94**	63.0	94.2	**83.85** (↑0.39)	**62.31** (↓8.54)
Energy [8]	81.68	89.97	86.01	72.68	90.13	45.25	82.39	66.13	59.72	95.11	79.99	73.83
Energy*	81.36	87.39	77.06	86.97	**94.15**	**28.3**	90.88	40.8	50.48	95.34	78.79 (↓1.20)	67.76 (↓6.07)
GEN [11]	**94.69**	**30.55**	86.2	74.58	60.77	99.33	67.15	97.46	83.51	79.1	78.46	76.20
GEN*	94.13	32.46	89.37	61.77	62.52	99.04	70.42	94.34	86.11	64.74	80.51 (↑2.05)	70.47 (↓5.73)
MCM [20]	93.25	45.23	86.15	77.22	62.58	98.57	69.57	96.22	84.12	79.55	79.13	79.36
MCM*	94.13	32.68	**90.06**	55.76	64.99	97.44	73.34	91.08	**86.65**	64.54	81.83(↑2.70)	68.30(↓11.06)
ResNet-101												
MSP [6]	93.12	34.72	71.32	88.07	44.25	99.16	63.26	92.98	81.1	68.21	70.61	76.63
MSP*	95.9	24.21	79.18	**75.6**	46.09	98.92	65.31	90.99	88.15	50.55	74.93 (↑4.32)	68.05 (↓8.58)
MaxLogit [7]	96.47	19.63	79.6	79.1	83.05	50.57	81.8	55.85	73.02	92.31	82.79	59.49
MaxLogit*	**98.76**	5.58	78.45	85.38	82.38	51.33	85.55	**45.96**	74.98	92.04	**84.02** (↑1.23)	**56.06** (↓3.43)
Energy [8]	89.9	56.88	76.44	85.98	**88.95**	**38.98**	82.98	57.87	60.29	96.44	79.71	67.23
Energy*	95.75	26.42	70.63	91.48	88.17	40.26	**86.64**	47.02	58.67	97.79	79.97 (↑0.26)	60.59 (↓6.64)
GEN [11]	98.17	9.8	71.5	89.41	39.66	99.99	59.47	98.42	83.09	69.36	70.38	73.40
GEN*	98.47	**5.24**	**82.2**	76.16	44.1	99.87	63.33	95.85	**91.59**	**45.47**	75.94 (↑5.56)	64.52 (↓8.88)
MCM [20]	96.13	25.33	72.41	90.17	41.08	99.83	61.81	96.72	83.11	69.36	70.91	76.28
MCM*	97.38	18.29	81.25	78.49	44.8	99.77	64.76	95.21	90.31	50.8	75.70(↑4.79)	68.51(↓7.77)

Table 1: *Per-Dataset Performance of OOD Detection Methods and the Ones Enhanced with TAG denoted with* $*$. The image encoders are ViT-L/14 and ResNet-101. The ID dataset is **CIFAR-100**. The number of augmentation $M = 10$ for TAG. The temperature $\tau = 0.01$ for all methods. Green indicates improvement and red indicates degradation.

the second block of Table 1. Particularly, MaxLogit [7] enhanced by TAG achieves the highest AUROC values and lowest FPR95 values on both ViT-L/14 and ResNet-101. The results of ImageNet-1k are shown in Table. 2. One can see that TAG again consistently improves the performance when using MSP [6], MaxLogit [7], and Energy [8] in terms of both AUROC and FPR95. When using GEN [11] as the OOD score, TAG is less effective on ImageNet-1k compared to CIFAR-100. We think this might be attributed to the limited capacity of pre-trained CLIP models. Specifically, the text prompt used in the training of CLIP is less informative, i.e., `a photo of` $\langle y_k \rangle$, where $\langle y_k \rangle$ is a noun and there is no other information such as activity information (i.e. verb) is provided. Moreover, the label information itself is quite restricted since there might be more than one object in the image [51].

OOD method	OpenImage-O		Textures		iNaturalist		ImageNet-O		Average	
	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓
ViT-L/14										
MSP [6]	89.85	41.76	83.13	59.11	91.52	37.14	80.74	65.65	86.31	50.92
MSP*	92.42	34.34	85.92	55.0	93.61	31.4	82.84	66.55	88.70 (↑2.39)	46.82 (↓4.10)
MaxLogit [7]	90.27	50.82	74.63	83.41	91.66	49.91	81.08	71.5	84.41	63.91
MaxLogit*	91.54	44.17	80.05	74.38	92.57	43.91	81.44	70.8	86.40 (↑1.99)	58.32 (↓5.59)
Energy [8]	87.31	67.56	69.63	89.63	88.48	68.57	78.89	76.8	81.08	75.64
Energy*	87.64	65.17	74.85	83.84	88.72	65.56	78.63	77.7	82.46 (↑1.38)	73.07 (↓2.57)
GEN [11]	**93.96**	**29.97**	87.48	53.59	**95.76**	**22.77**	**84.75**	**62.95**	**90.49**	**42.32**
GEN*	93.72	31.68	86.85	55.85	94.79	28.37	84.33	67.35	89.92 (↓0.57)	45.81 (↑3.49)
MCM [20]	93.08	35.04	86.62	55.66	94.96	28.3	82.59	68.55	89.31	46.89
MCM*	93.05	36.96	**88.55**	**52.02**	93.9	37.22	82.04	73.8	89.39 (↑0.08)	50.00 (↑3.11)
ResNet-101										
MSP [6]	83.53	60.68	79.36	66.94	82.35	61.86	70.47	82.4	78.93	67.97
MSP*	85.39	59.03	82.62	61.24	85.61	58.88	71.72	84.4	81.34 (↑2.41)	65.89 (↓2.08)
MaxLogit [7]	83.94	72.86	69.61	91.96	82.33	82.9	71.78	86.05	76.91	83.44
MaxLogit*	84.69	72.62	75.47	88.53	83.24	79.85	72.08	86.7	78.87 (↑1.96)	81.92 (↓1.52)
Energy [8]	79.56	85.26	62.19	97.23	77.53	94.16	69.36	87.75	72.16	91.10
Energy*	79.2	84.13	67.19	95.27	77.11	92.32	69.15	89.35	73.16 (↑1.00)	90.27 (↓0.83)
GEN [11]	**89.24**	**52.86**	84.99	62.46	89.58	**53.15**	**77.23**	**82.25**	85.26	62.68
GEN*	88.48	55.89	85.01	65.33	89.12	57.53	76.31	84.15	84.73 (↓0.53)	65.72 (↑3.04)
MCM [20]	88.82	54.82	86.26	59.28	**89.93**	53.35	75.15	83.6	85.04	62.76
MCM*	88.38	56.27	**88.25**	**51.53**	89.21	57.12	75.36	84.3	**85.30** (↑0.26)	**62.31** (↓0.45)

Table 2: *Per-Dataset Performance of OOD Detection Methods and the Ones Enhanced with TAG denoted with* $*$. The image encoders are ViT-L/14 and ResNet-101. The ID dataset is **ImageNet-1k**. The number of augmentation $M = 10$ for TAG. The temperature $\tau = 0.01$ for all methods except for MCM [20]. Green indicates improvement and red indicates degradation.

Averaged results on other architectures

To further investigate the effectiveness and robustness of TAG, we conducted OOD detection on three more models including two ViT-based models, which are ViT-B/16 and ViT-B/32, and one more ResNet-based model, ResNet-50. The performance is evaluated on both CIFAR-100 and ImageNet-1k. The results of CIFAR-100 are averaged over 5 different OOD datasets and shown in the top half of the Table. 3. It is undoubted that TAG again substantially and constantly improves the performance of all baseline score functions across 5 datasets and 3 architectures on CIFAR-100. Specifically, one can see that MaxLogit [7] enhanced by TAG achieves the best performance in terms of AUROC on average and GEN [11] enhanced by TAG obtains the lowest FPR95 values. For ImageNet-1k, the averages are calculated with 4 OOD datasets

	OOD Method	ViT-B/16		ViT-B/32		ResNet-50		Average	
		AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓
CIFAR-100	MSP [6]	75.05	85.30	77.05	75.45	60.25	90.01	70.78	83.59
	MSP*	79.21	69.48	78.45	74.20	71.55	**62.55**	76.40 (↑5.62)	68.74 (↓14.85)
	MaxLogit [7]	74.70	85.30	83.56	66.59	50.74	92.22	69.67	81.37
	MaxLogit*	**83.86**	67.32	**87.03**	**59.84**	**75.33**	78.57	**82.07** (↑12.40)	68.58 (↓12.79)
	Energy [8]	69.64	86.62	80.05	71.51	45.27	93.20	64.99	83.78
	Energy*	79.31	71.45	84.07	64.56	69.81	85.75	77.73 (↑12.74)	73.92 (↓9.86)
	GEN [11]	75.07	82.14	77.98	66.36	60.38	84.90	71.14	77.80
	GEN*	81.38	**62.81**	78.23	68.31	71.89	63.28	77.17 (↑6.03)	**64.80** (↓13.00)
	MCM [20]	75.55	84.76	77.93	73.00	60.12	88.30	71.2	82.02
	MCM*	80.85	65.75	78.62	75.06	71.93	63.33	77.13 (↑5.93)	68.05 (↓13.97)
ImageNet-1k	MSP [6]	82.85	59.36	79.79	65.00	79.22	67.41	80.62	63.92
	MSP*	85.13	57.76	82.03	64.63	81.10	65.33	82.75 (↑2.13)	62.57 (↓1.35)
	MaxLogit [7]	82.84	68.00	80.03	72.35	78.34	80.11	80.40	73.49
	MaxLogit*	84.48	65.92	82.54	67.98	79.09	79.90	82.03 (↑1.63)	71.26 (↓2.23)
	Energy [8]	79.26	79.09	76.48	82.03	74.11	88.66	76.61	83.26
	Energy*	80.23	79.92	78.73	78.48	74.16	88.73	77.71 (↑1.10)	82.38 (↓0.88)
	GEN [11]	**88.70**	**50.09**	**86.64**	56.34	86.02	59.19	**87.12**	55.21
	GEN*	87.83	54.95	85.64	62.65	84.46	65.53	85.98 (↓1.14)	61.04 (↑5.83)
	MCM [20]	88.18	51.9	86.31	**55.45**	**86.09**	**57.17**	86.86	**54.83**
	MCM*	87.72	56.94	86.31	59.08	85.54	62.02	86.52 (↓0.34)	59.34 (↑4.51)

Table 3: *Averaged Performance of Various OOD Detection Methods and the Ones Enhanced by TAG denoted with* $*$. Results are shown for ViT-B/16, ViT-B/32, and ResNet-50. For CIFAR-100, averages are computed across 5 OOD datasets, while for ImageNet-1k, the averages are derived from 4 OOD datasets. Green indicates improvement and red indicates degradation.

and shown in the bottom half of the Table. 3. TAG continually boosts the performance of OOD detection using MSP [6], MaxLogit [7] and Energy [8]. Additionally, the score function GEN [11] devised for the discriminative classifier achieves the best AUROC values and MCM [20] obtains the smallest FPR95 values on ImageNet-1k. In short, applying TAG on top of different score functions generally is a good idea to boost the performance fo OOD detection. Detailed results for each architecture can be found in supplementary material.

4.2 Ablation studies

Analysis of text embeddings

We observe that the improvement on ImageNet-1k is less pronounced than CIFAR-100. The hypothesis is that the pre-trained text embeddings for each

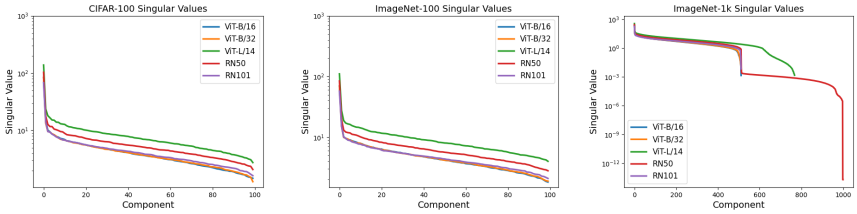

Figure 4: *Singular Values Visualization.* From left to right, the datasets include CIFAR-100 [41], ImageNet-100, and ImageNet-1k [42].

class are not separable well. We confirm this by computing the rank of concatenated text embeddings and the visualization of singular values for CIFAR-100, ImageNet-100 and ImageNet-1k is shown in Fig. 4. One can see that the rank is 100 for both CIFAR-100 and ImageNet-100 across 5 different models. While the rank of the concatenated text embeddings for ImageNet-1k is generally less than 710 and most singular values are quite small. Detailed rank information with different models can be found in the supplementary material. We suspect that this is due to our utilized text prompts not covering the entire semantic space. Therefore we perform OOD detection on ImageNet-100, which is a subset of ImageNet-1k with 100 classes and the data list is provided by MCM [20]. The corresponding results can be found in Table. 4, and it is apparent that TAG consistently improves the baseline methods. MCM [20] combined with TAG is leading in terms of both AUROC and FPR95.

Choice of τ and M

We empirically show the performance gap between the baseline methods and the ones enhanced with TAG using different temperatures τ and the number of text prompt augmentations M in terms of both AUROC and FPR95. Experiments of using different τ with CIFAR-100 as the ID dataset are conducted on ViT-B/16 and are presented in Figure. 5, in which each column represents one score function. The first row represents the results of regarding AUROC, and the second row indicates FPR95 performance. It is shown in Figure. 5 that TAG (with $M = 10$) could persistently improve the performance of the baseline OOD score in terms both of AUROC and FPR95 except for GEN [11] with $\tau = 0.1$. The evaluation regarding temperature τ for other architecture

OOD Method	OpenImage-O		Texture		iNaturalist		ImageNet-O		Average	
	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓
MSP [6]	94.12	34.34	90.69	50.89	95.52	29.3	90.01	50.2	92.58	41.18
MSP*	95.55	**26.41**	92.65	43.97	96.31	**21.74**	**91.35**	46.3	93.97 (↑1.39)	**34.60** (↓6.58)
MaxLogit [7]	93.97	38.39	83.83	75.14	94.95	34.93	90.58	51.95	90.83	50.10
MaxLogit*	94.99	30.29	88.79	58.39	95.79	25.28	91.31	**45.6**	92.72 (↑1.89)	39.89 (↓10.21)
Energy [8]	92.55	48.74	81.14	80.23	93.5	44.9	89.5	56.4	89.17	57.57
Energy*	93.11	43.28	86.12	66.59	94.17	36.71	89.86	51.85	90.82 (↑1.65)	49.61 (↓7.96)
GEN [11]	95.21	30.75	91.11	49.96	**96.47**	23.94	90.58	54.65	93.34	39.83
GEN*	95.3	31.46	94.02	**37.34**	95.81	30.51	90.64	54.7	93.94 (↑0.60)	38.50 (↓1.33)
MCM [20]	95.36	30.58	91.4	50.06	**96.6**	23.92	90.87	52.75	93.56	39.33
MCM*	**95.64**	28.22	**94.06**	38.39	96.2	26.17	91.1	51.45	**94.25** (↑0.69)	36.06 (↓3.27)

Table 4: *Per-Dataset Performance of OOD Detection Methods and the Ones Enhanced with TAG denoted with *.* The image encoders are ViT-L/14. The ID dataset is **ImageNet-100**. Green indicates improvement and red indicates degradation.

can be found in the supplementary material.

Additionally, we also investigate the effect of using different numbers of text prompt augmentations, and the results (with $\tau = 0.01$) on CIFAR-100 and ImageNet-1k are presented in Figure. 6. One can see that it is adequate to set $M = 2$ for CIFAR-100 as the ID dataset and $M = 10$ for ImageNet-1k as the ID dataset. Results on other architectures can be found in the supplementary material.

Combining with DCLIP [37] and WaffleCLIP [38]

We combine TAG with the default text prompt extended with descriptors generated by GPT-3 denoted by DCLIP [37] and prolonged with random characters or words denoted by WaffleCLIP [38]. The generated descriptors for each class are provided by WaffleCLIP [38]. CLIP means the default prompt `a photo of a` $\langle y_k \rangle$ is utilized. The OOD score is MSP with $\tau = 0.01$. One can see that TAG could further enhance the performance of OOD detection under various descriptors. WaffleCLIP [38] enhanced by TAG is leading in terms of AUROC. Results on other architectures with different score functions can be found in the supplementary material.

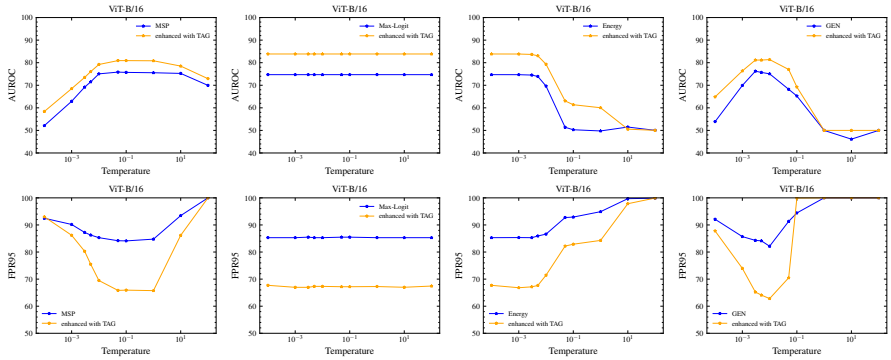

Figure 5: *Averaged Performance (over 5 OOD Datasets) of TAG Applied with Different Temperature τ.* TAG performance in terms of AUROC values (top row) and FPR95 (bottom row). Each column denotes different score functions including MSP [6], MaxLogit [7], Energy [8], and GEN [11] (from left to right).

5 Conclusion and Discussions

In this work we explore the benefits of adapting OOD scores designed for discriminative classifiers (e.g. trained with the cross-entropy loss) to vision-language models (i.e. CLIP [17] trained with an InfoNCE [40] loss). Models like CLIP [17] enable the use of various OOD scores to perform zero-shot OOD detection by only accessing the label information of the ID dataset, and they also allow variability in the resulting OOD scores by varying the text prompts. Our proposed method named TAG (Text prompt AuGmentation)

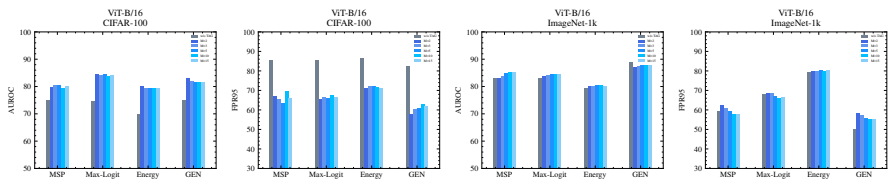

Figure 6: *Averaged Performance of TAG Varying with Different Augmentations M.* The left two column corresponds to CIFAR-100 [41] dataset, and the right two columns corresponds to ImageNet-1k [42].

Prompt	OpenImage-O		Textures		iNaturalist		ImageNet-O		Average	
	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓
ViT-B/16										
CLIP [17]	86.39	51.51	81.57	61.12	87.53	49.66	75.92	75.15	82.85	59.36
CLIP*	89.18	45.71	84.13	60.04	89.69	48.09	77.52	77.2	85.13 (↑2.28)	57.76 (↓1.60)
DCLIP [37]	81.87	62.53	78.05	69.09	80.68	61.96	72.68	79.15	78.32	68.18
DCLIP*	86.3	57.91	83.26	63.84	84.4	70.26	75.4	81.7	82.34 (↑4.02)	68.43 (↑0.25)
WaffleCLIP [38]	83.19	59.88	79.89	67.42	82.49	61.04	75.0	75.9	80.14	66.06
WaffleCLIP*	88.72	47.78	85.63	55.48	87.46	55.67	78.82	74.4	85.16 (↑5.02)	58.33 (↓7.73)

Table 5: *Performance of using different descriptors with* $M = 10$ *and* $\tau = 0.01$. The architecture is ViT-B/16. The ID dataset is ImageNet-1k [42]. * denotes the methods enhanced by TAG. The score function is MSP. Green indicates improvement and red indicates degradation.

leverages this variability, is easy to implement and effective for various OOD scores across different architectures with the minimal knowledge. It does not rely on the external knowledge from LLMs with the risk of hallucination or prompt ensembling. TAG offers significant improvements on standard OOD scores for most tested network models and datasets. A focus of future work is the less pronounced improvement on ImageNet-1k, which is likely to be attributed to the (simple) text prompts not exhausting CLIP's latent space, but may also be related to intrinsic shortcomings of the InfoNCE loss [52].

6 Supplementary Material

I Softmax Temperature and Tuning the FPR

In this section we demonstrate that MaxLogit can always be outperformed by MSP/MCM in terms of the FPR by choosing the right temperature. Let $s^{(1)}(\mathbf{x}), \dots, s^{(K)}(\mathbf{x})$ be the logits for a sample $\mathbf{x}$ in descending sort order. The MaxLogit decision rule is given by

$$\begin{cases} 1 & \text{if } s^{(1)}(\mathbf{x}) \geq \lambda^0 \\ 0 & \text{if } s^{(1)}(\mathbf{x}) < \lambda^0, \end{cases} \tag{E.5}$$

where λ^0 is a score threshold to achieve a certain TPR (e.g. 95%). Let $\text{FPR}^0(\lambda^0)$ be the FPR of MaxLogit for a given choice of λ^0.

It is beneficial to state the MSP/MCM decision rule using log-probabilities,

$$\begin{cases} 1 & \text{if } s^{(1)}(\mathbf{x})/\tau - \text{LSE}_i(s^{(i)}/\tau) \geq \lambda \\ 0 & \text{if } s^{(1)}(\mathbf{x})/\tau - \text{LSE}_i(s^{(i)}/\tau) < \lambda, \end{cases} \tag{E.6}$$

where the log-sum-exp is given by

$$\text{LSE}_i(a_i) := \log \sum\nolimits_{i=1}^{K} \exp(a_i). \tag{E.7}$$

Here i denotes the class index, a_i is the corresponding logit, and K is the total number of classes per dataset. Further, let $Q_{\mathbf{x}}$ be the distribution of outliers. As in the main text we assume that $s^{(i)}(\mathbf{x})$ are bounded from below, more precisely we assume that $s^{(i)}(x) \geq L$ $Q_{\mathbf{x}}$-a.e. (i.e. $Q_{\mathbf{x}}(\min_i s^{(i)}(\mathbf{x}) \geq L) = 1$). Hence, w.l.o.g. we can assume that $\lambda^0 > L$. Now

$$\begin{aligned} \text{FPR}(\lambda, \tau) &= Q_{\mathbf{x}}\left(\tfrac{1}{\tau} s^{(1)}(\mathbf{x}) - \text{LSE}_i(\tfrac{1}{\tau} s^{(i)}) \geq \lambda\right) \\ &= Q_{\mathbf{x}}\left(s^{(1)}(\mathbf{x}) \geq \tau\lambda + \tau \text{LSE}_i(\tfrac{1}{\tau} s^{(i)}))\right) \\ &\leq Q_{\mathbf{x}}\left(s^{(1)}(\mathbf{x}) \geq \tau\lambda + \tau \text{LSE}_i(\tfrac{1}{\tau} L))\right) \\ &= Q_{\mathbf{x}}\left(s^{(1)}(\mathbf{x}) \geq \tau\lambda + \tau \log(K \exp(L/\tau))\right) \\ &= Q_{\mathbf{x}}\left(s^{(1)}(\mathbf{x}) \geq \tau(\lambda + \log K) + L\right). \end{aligned} \tag{E.8}$$

If we choose $\tau > 0$ and λ such that $\tau(\lambda + \log K) + L \geq \lambda^0$, then

$$Q_{\mathbf{x}}\left(s^{(1)}(\mathbf{x}) \geq \tau(\lambda + \log K) + L\right) \leq Q_{\mathbf{x}}\left(s^{(1)}(\mathbf{x}) \geq \lambda^0\right) = \mathrm{FPR}^0(\lambda^0). \tag{E.9}$$

If we fix $\lambda > -\log K$ (implying $e^{\lambda} > 1/K$ in probability space), then the temperature τ has to satisfy

$$\tau \geq \frac{\lambda^0 - L}{\lambda + \log K}. \tag{E.10}$$

Since both numerator and denominator are positive, $\tau > 0$ has therefore to be sufficiently large to obtain $\mathrm{FPR}(\lambda, \tau) < \mathrm{FPR}^0(\lambda^0)$.

II Datasets

Although performing OOD detection in ImageNet-1k is more challenging, it is empirically shown in [53] that there is no single OOD score function can consistently outperform others across all benchmarks. Therefore, it is necessary to perform OOD detection on a small-scale dataset with CIFAR- 100 [41] as the ID dataset and a more realistic large-scale dataset with ImageNet-1k as the ID dataset. One extra OOD benchmark with ImageNet-100 as ID data is also utilized. We use the curated ImageNet-100 from MCM [20] and the script for constructing the dataset and the corresponding class list can be found at https://github.com/deeplearning-wisc/MCM.

The small-scale OOD benchmark incorporates 5 datasets, which are SVHN [43], iSUN [44], Textures [46], LSUN [47] and Places365 [45]. The large-scale OOD benchmark consists of 4 datasets, which are OpenImage-O [49], Textures [46], iNaturalist [50] and ImageNet-O [48]. The detailed information for aforementioned OOD datasets is summarized in Table 6.

III Text Embedding Analysis

There are five models including ViT-based and ResNet-based models utilized to demonstrate the effectiveness of TAG. Moreover, three datasets including CIFAR-100 [41], ImageNet-100 [42], and ImageNet-1k [42] are used as ID datasets. Assuming K semantic labels of the ID dataset, a matrix $W \in \mathbb{R}^{d \times K}$ is simply constructed by stacking the d-dimensional text embedding (column

Dataset	Image distribution	# Images
SVHN [43]	predefined (OOD) class list	26,032
iSUN [44]	predefined (OOD) class list	8,925
Places365 [45]	predefined (OOD) class list	10,000
LSUN [47]	predefined (OOD) class list	2,000
Textures [46]	predefined (OOD) class list	5,640
ImageNet-O [48]	natural adversarial images	2,000
OpenImage-O [10]	natural (OOD) class distribution	17,632
iNaturalist [50]	predefined (OOD) class list	10,000

Table 6: *Specifications of OOD datasets.*

vectors) of each class coming from the text encoder. Afterward, the rank information with the corresponding singular value is obtained via `svd`(W). The rank information of concatenated text embeddings for each ID dataset with different models is summarized in Table **??**. One can see that the rank of different models for ImageNet-1k is constantly smaller than 1000, which implies that the learned text embeddings for each class on ImageNet-1k are even less separable.

Model	**Text embedding**	**Rank of text embeddings on**		
		CIFAR100	ImageNet-100	ImageNet-1k
ViT-B/16	512	100	100	509
ViT-B/32	512	100	100	509
ViT-L/14	768	100	100	708
ResNet-50	1024	100	100	512
ResNet-101	512	100	100	510

Table 7: *Specifications of Different Models with Different Datasets*: dimensionality of the text embedding space and the rank of the concatenated text embeddings based on the class names within each ID dataset over different models.

IV DCLIP [37] and WaffleCLIP [38]

DCLIP [37] harnesses knowledge from large language models (LLMs) to generate the description for each class in order to improve the zero-shot classification performance. Instead, WaffleCLIP [38] argues it is possible to replace the meaningful descriptors generated by LLMs with random words or characters

without diminishing the zero-shot classification accuracy.

We further investigate whether TAG can be applied to the default prompt extended with descriptors generated by DCLIP [37] and WaffleCLIP [38]. The corresponding descriptors are provided by WaffleCLIP [38] and can be found at https://github.com/ExplainableML/WaffleCLIP. The minimum number of descriptors for each class is 2; hence, 2 descriptors per class are selected when experimenting with DCLIP [37]. As for WaffleCLIP [38], all classes utilize the same descriptors with the size of 10. Due to the limited computing resources, the maximum text augmentation $M = 10$. To be specific, the text augmentation $M = 5$ when using WaffleCLIP [38] on all architectures except for ViT-B/16. The OOD results using MSP score on other architectures are shown in Table 8. It can be seen TAG could further improve the OOD results on most datasets and architectures when using MSP score.

V Comparison with Prompt Ensemble

We further compare TAG with prompt ensembling. We employ the prompt templates provided by CLIP with the size of 80. The ensemble are obtained by averaging their OOD score. The averaged results on ImageNet-100 with two types of architectures are shown in Table 9. It is worthwhile to note that TAG could further boost the performance using prompt ensemble equipped with various OOD scores.

VI Detailed OOD Detection Performance Results

In this section, we provide the detailed results of each ID dataset including CIFAR-100, ImageNet-100, and ImageNet-1k across 5 different models consisting of ViT-B/16, ViT-B/32, ViT-L/14, ResNet-50, and ResNet-101. The corresponding results for each dataset with different models can be found in Table 10, Table **??**, and Table **??**, respectively. $\tau = 0.01$ for all experiments except for MCM [20] (where $\tau = 1$). One can see that, in general, TAG boosts the performance of various baseline score functions over different datasets across different architectures. Notably, TAG significantly enhances the performance of OOD detection in terms of FPR95 on the CIFAR-100 dataset compared to ImageNet-100 and ImageNet-1k. Moreover, the improvement on ImageNet-100 is slightly superior to that on ImageNet-1k, except for the model ResNet-101.

VII Extended Results for Effective τ and M

In this section, more detailed results of using different temperature τ and different numbers of augmentations M on different architectures and datasets are reported. First, the performance varying with different temperatures $\tau = \{0.0001, 0.001, 0.003, 0.005, 0.01, 0.05, 0.1, 1, 10, 100\}$ is presented. The evaluation performed on ViT-B/32, ViT-L/14, ResNet-50, and ResNet-101 for CIFAR-100 and ImageNet-1k are presented in Fig. 7, 8 and 9, 10, 13, respectively. In addition, the results for ImageNet-100 can be found in Fig. 11 and Fig. 12. One can see that TAG generally improves performance with a noticeable margin in terms of both AUROC and FPR95 under different temperature values τ in CIFAR-100 and ImageNet-100 across different architectures. Again, the performance gain is less pronounced on ImageNet-1k.

Subsequently, the results of using different augmentations $M = \{2, 3, 5, 10, 15\}$ with $\tau = 0.01$ are presented in Fig. 15. The left two columns correspond to the results on CIFAR-100 in terms of AUROC and FPR95. The right two columns correspond to ImageNet-1k. Additionally, the evaluation performed on ImageNet-100 is presented in Fig. 14. In general, more augmentations are implemented, resulting in enhanced performance regarding both AUROC and FPR95. Particularly, more augmentations applied to the text prompt tend to obtain a lower FPR95 value in ImageNet-100. Overall, setting $M = 10$ in general is a satisfactory choice.

Prompt	OpenImage-O		Textures		iNaturalist		ImageNet-O		Average	
	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓
ViT-B/16										
CLIP	86.39	51.51	81.57	61.12	87.53	49.66	75.92	75.15	82.85	59.36
CLIP*	89.18	45.71	84.13	60.04	89.69	48.09	77.52	77.2	85.13 (↑2.28)	57.76 (↓1.60)
DCLIP [37]	81.87	62.53	78.05	69.09	80.68	61.96	72.68	79.15	78.32	68.18
DCLIP*	86.3	57.91	83.26	63.84	84.4	70.26	75.4	81.7	82.34 (↑4.02)	68.43 (↑0.25)
WaffleCLIP [38]	83.19	59.88	79.89	67.42	82.49	61.04	75.0	75.9	80.14	66.06
WaffleCLIP*	88.72	47.78	85.63	55.48	87.46	55.67	78.82	74.4	85.16 (↑5.02)	58.33 (↓7.73)
ViT-B/32										
CLIP	83.89	58.71	79.12	67.05	85.57	53.92	70.6	80.3	79.79	65.00
CLIP*	86.3	56.09	81.47	65.91	87.4	54.67	72.94	81.85	82.03 (↑2.24)	64.63 (↓0.37)
DCLIP [37]	81.44	63.91	78.28	70.66	80.43	60.4	70.34	83.5	77.62	69.62
DCLIP*	84.35	62.3	81.45	66.94	82.76	63.23	73.29	83.8	80.46 (↑2.84)	69.07 (↓0.55)
WaffleCLIP [38]	83.21	62.58	80.35	65.7	82.77	63.6	71.81	80.9	79.53	68.19
WaffleCLIP*	86.63	55.85	82.95	59.84	88.08	54.52	74.74	80.6	83.10 (↑3.57)	62.70 (↓5.49)
ViT-L/14										
CLIP	89.85	41.76	83.13	59.11	91.52	37.14	80.74	65.65	86.31	50.92
CLIP*	92.42	34.34	85.92	55.0	93.61	31.4	82.84	66.55	88.70 (↑2.39)	46.82 (↓4.10)
DCLIP [37]	88.95	45.36	82.13	61.96	89.77	43.48	80.46	69.35	85.33	55.04
DCLIP*	90.8	43.51	84.65	63.64	88.91	58.16	82.24	72.7	86.65 (↑0.32)	59.50 (↑4.46)
WaffleCLIP [38]	89.97	41.35	83.97	56.18	91.05	38.68	81.28	67.05	86.57	50.81
WaffleCLIP*	91.71	37.79	86.49	53.68	90.81	46.42	83.18	67.95	88.05 (↑1.48)	51.46 (↑0.65)
RN50										
CLIP	82.58	62.28	79.87	65.6	85.99	55.16	68.42	86.6	79.22	67.41
CLIP*	84.7	58.86	81.79	62.81	87.82	53.16	70.1	86.5	81.10 (↑1.88)	65.33 (↓2.08)
DCLIP [37]	79.26	68.62	77.28	72.42	79.05	64.61	66.87	86.1	75.62	72.94
DCLIP*	83.36	60.75	82.13	64.48	82.24	61.89	70.17	86.5	79.48 (↑3.86)	68.41 (↓4.53)
WaffleCLIP [38]	81.3	65.77	80.19	67.03	83.23	59.28	68.25	86.7	78.24	69.70
WaffleCLIP*	84.7	56.48	82.67	61.05	86.41	55.31	71.06	84.95	81.21 (↑2.97)	64.45 (↓5.25)
RN101										
CLIP	83.53	60.68	79.36	66.94	82.35	61.86	70.47	82.4	78.93	67.97
CLIP*	85.39	59.03	82.62	61.24	85.61	58.88	71.72	84.4	81.34 (↑2.41)	65.89 (↓2.08)
DCLIP [37]	81.88	62.41	79.48	69.4	79.91	62.43	69.85	83.75	77.78	69.50
DCLIP*	84.17	62.0	81.26	68.64	82.2	63.27	72.25	84.7	79.97 (↑2.19)	69.65 (↑0.15)
WaffleCLIP [38]	83.96	59.38	81.49	64.11	82.36	60.52	70.74	81.7	79.64	66.43
WaffleCLIP*	85.38	58.33	83.78	60.85	84.94	62.81	72.48	84.1	81.64 (↑2.00)	66.52 (↑0.09)

Table 8: *Performance of using different descriptors with MSP ($\tau = 0.01$) as score.* The ID dataset is ImageNet-1k [42]. * denotes the methods enhanced by TAG. The score function is **MSP**. Green indicates improvement and red indicates degradation.

OOD method	ViT-B/32		RN50	
	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓
MSP†	91.20	41.31	90.77	43.92
MSP*	92.24 (↑1.04)	38.06 (↓3.25)	91.63 (↑0.86)	40.41 (↓3.51)
MaxLogit†	88.88	52.69	87.61	59.83
MaxLogit*	89.47 (↑0.59)	51.28(↓1.41)	86.92 (↓0.69)	59.42 (↓0.41)
Energy†	86.38	60.13	84.62	69.84
Energy*	86.62 (↑0.24)	58.37 (↓1.76)	83.28 (↓1.34)	69.25 (↓0.59)
GEN†	92.81	40.22	92.85	37.48
GEN*	92.87 (↑0.06)	41.34 (↑1.12)	92.98 (↑0.13)	37.46 (↓0.02)
MCM†	92.94	40.18	92.89	40.23
MCM*	93.12 (↑0.18)	37.53 (↓2.65)	92.98 (↑0.09)	36.46 (↓3.77)

Table 9: *Comparison with prompt ensemble.* The ID dataset is ImageNet-100, the results are averaged over 4 OOD datasets. OOD methods using prompt ensemble and further enhanced with TAG denoted with † and *, respectively. Green/red indicates improvement/degradation.

Models + OOD Method	SVHN		iSUN		Places365		Textures		LSUN		Average	
	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓
ViT-B/16												
MSP [6]	84.7	80.02	77.58	87.37	57.43	98.56	70.16	93.0	85.4	67.56	75.05	85.30
MSP*	90.28	53.3	81.7	82.34	58.61	95.28	73.54	80.23	91.94	36.24	79.21 (↑4.16)	69.48 (↓15.82)
MaxLogit [7]	79.72	93.37	86.71	66.71	67.81	85.06	67.2	88.78	72.07	92.57	74.70	85.30
MaxLogit*	88.65	70.19	86.15	69.93	84.74	51.88	84.52	52.29	75.22	92.32	83.86 (↑9.16)	67.32 (↓17.98)
Energy [8]	71.11	97.45	85.89	70.6	69.66	80.99	62.79	88.33	58.75	95.73	69.64	86.62
Energy*	82.02	83.08	83.08	77.61	87.69	44.52	83.39	55.25	60.37	96.78	79.31 (↑9.67)	71.45 (↓15.17)
GEN [11]	90.89	58.64	79.44	89.3	52.51	99.85	64.65	97.8	87.87	65.09	75.07	82.14
GEN*	96.73	17.67	82.85	81.29	59.15	97.84	74.86	82.43	93.29	34.84	81.38 (↑6.31)	62.81 (↓19.33)
MCM [20]	88.7	73.68	79.78	87.85	54.23	99.56	67.49	96.47	87.57	66.22	75.55	84.76
MCM*	94.6	32.77	82.8	82.44	58.97	97.3	74.78	82.23	93.09	34.01	80.85 (↑5.30)	65.75 (↓19.01)
ViT-B/32												
MSP [6]	94.27	33.81	77.44	85.82	56.17	98.78	70.43	92.57	86.93	66.29	77.05	75.45
MSP*	94.63	34.48	78.89	91.19	58.05	98.56	70.35	90.92	90.33	55.86	78.45 (↑1.4)	74.20 (↓1.25)
MaxLogit [7]	90.95	55.81	89.51	56.77	76.89	71.72	71.79	82.02	88.64	66.64	83.56	66.59
MaxLogit*	93.52	43.3	85.9	69.46	85.08	53.05	82.71	60.16	87.96	73.22	87.03 (↑3.47)	59.84 (↓6.75)
Energy [8]	80.12	75.24	88.85	59.18	80.26	62.21	68.25	81.99	82.79	78.94	80.05	71.51
Energy*	86.6	63.95	83.3	73.86	88.71	40.5	82.31	58.14	79.43	86.37	84.07 (↑4.02)	64.56 (↓6.95)
GEN [11]	98.44	8.11	82.55	83.79	51.71	99.94	64.25	97.22	92.94	42.72	77.98	66.36
GEN*	97.03	19.31	82.07	85.88	51.9	99.91	66.53	94.77	93.61	41.7	78.23 (↑0.25)	68.31 (↑1.95)
MCM [20]	96.79	21.63	81.21	86.82	53.49	99.72	67.47	96.45	90.69	60.38	77.93	73.00
MCM*	96.22	31.83	81.31	91.64	54.39	99.83	68.76	94.88	92.4	57.13	78.62 (↑0.69)	75.06 (↑2.06)
ViT-L/14												
MSP [6]	90.54	51.91	84.23	75.26	65.52	95.99	72.11	91.37	83.03	75.71	79.09	78.05
MSP*	92.91	34.31	89.77	50.94	69.47	90.08	77.19	80.6	86.27	64.34	83.12 (↑4.03)	64.05 (↓14.0)
MaxLogit [7]	88.62	69.28	82.97	80.32	92.39	33.97	91.01	38.09	62.72	93.97	83.46	70.85
MaxLogit*	88.17	69.34	84.06	77.8	92.88	32.29	91.16	37.94	63.0	94.2	83.85 (↑0.39)	62.31 (↓8.54)
Energy [8]	81.68	89.97	86.01	72.68	90.13	45.25	82.39	66.13	59.72	95.11	79.99	73.83
Energy*	81.36	87.39	77.06	86.97	94.15	28.3	90.88	40.8	50.48	95.34	78.79 (↓1.20)	67.76 (↓6.07)
GEN [11]	94.69	30.55	86.2	74.58	60.77	99.33	67.15	97.46	83.51	79.1	78.46	76.20
GEN*	94.13	32.46	89.37	61.77	62.52	99.04	70.42	94.34	86.11	64.74	80.51 (↑2.05)	70.47 (↓5.73)
MCM [20]	93.25	45.23	86.15	77.22	62.58	98.57	69.57	96.22	84.12	79.55	79.13	79.36
MCM*	94.13	32.68	90.06	55.7	64.99	97.44	73.34	91.08	86.65	64.54	81.83 (↑2.7)	68.30 (↓11.06)
ResNet-50												
MSP [6]	82.9	74.15	68.52	88.37	29.17	99.87	49.12	99.34	71.52	88.31	60.25	90.01
MSP*	98.65	5.74	62.42	96.35	40.18	95.99	65.48	76.81	91.01	37.86	71.55 (↑11.30)	62.55 (↓27.46)
MaxLogit [7]	61.99	98.35	82.24	66.98	29.31	98.12	26.93	99.26	53.22	98.4	50.74	92.22
MaxLogit*	89.11	66.5	79.8	79.03	64.47	78.11	70.45	75.23	72.82	93.97	75.33 (↑24.59)	78.57 (↓13.65)
Energy [8]	45.85	99.71	81.97	71.41	33.34	96.79	23.06	99.01	42.13	99.07	45.27	93.20
Energy*	72.35	95.28	79.7	80.21	70.82	74.61	68.25	80.21	57.94	98.42	69.81 (↑24.54)	85.75 (↓7.45)
GEN [11]	91.31	53.81	71.12	88.69	21.03	99.99	38.24	99.75	80.2	82.25	60.38	84.90
GEN*	99.15	3.3	64.85	95.44	36.3	99.71	65.84	85.32	93.32	32.64	71.89 (↑11.51)	63.28 (↓21.62)
MCM [20]	87.35	67.67	70.51	87.81	23.41	99.99	43.19	99.65	76.15	86.4	60.12	88.30
MCM*	99.12	3.6	64.42	96.45	37.21	99.0	66.16	82.3	92.72	35.29	71.93 (↑11.81)	63.33 (↓24.97)
ResNet-101												
MSP [6]	93.12	34.72	71.32	88.07	44.25	99.16	63.26	92.98	81.1	68.21	70.61	76.63
MSP*	95.9	24.21	79.18	75.6	46.09	98.92	65.31	90.99	88.15	50.55	74.93 (↑4.32)	68.05 (↓8.58)
MaxLogit [7]	96.47	19.63	79.6	79.1	83.05	50.57	81.8	55.85	73.02	92.31	82.79	59.49
MaxLogit*	98.76	5.58	78.45	85.38	82.38	51.33	85.55	45.96	74.98	92.04	84.02 (↑1.23)	56.06 (↓3.43)
Energy [8]	89.9	56.88	76.44	85.98	88.95	38.98	82.98	57.87	60.29	96.44	79.71	67.23
Energy*	95.75	26.42	70.63	91.48	88.17	40.26	86.64	47.02	58.67	97.79	79.97 (↑0.26)	60.59 (↓6.64)
GEN [11]	98.17	9.8	71.5	89.41	39.66	99.99	59.47	98.42	83.09	69.36	70.38	73.40
GEN*	98.47	5.24	82.2	76.16	44.1	99.87	63.33	95.85	91.59	45.47	75.94 (↑5.56)	64.52 (↓8.88)
MCM [20]	96.13	25.33	72.41	90.17	41.08	99.83	61.81	96.72	83.11	69.36	70.91	76.28
MCM*	97.38	18.29	81.25	78.49	44.8	99.77	64.76	95.21	90.31	50.8	75.70 (↑4.79)	68.51 (↓7.77)

Table 10: *Per-Dataset Performance of Various OOD Methods and the Ones Enhanced with TAG.* We set $M = 10$ (and $\tau = 0.01$ for MSP, Energy and GEN). The ID dataset is **CIFAR-100** [41]. Green indicates improvement and red indicates degradation.

Models + OOD Method	OpenImage-O		Texture		iNaturalist		ImageNet-O		Average	
	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓
ViT-B/16										
MSP [6]	93.2	36.66	90.32	46.51	94.21	34.08	87.93	54.1	91.41	42.84
MSP*	94.46	29.47	91.66	47.0	93.99	33.3	89.51	53.05	92.41 (↑1.00)	40.70 (↓2.14)
MaxLogit [7]	93.56	37.36	84.61	71.38	94.44	35.39	89.05	53.45	90.42	49.39
MaxLogit*	94.14	33.59	87.85	63.55	93.67	37.36	89.67	51.45	91.33 (↑0.91)	46.49 (↓2.90)
Energy [8]	92.0	46.56	81.27	79.22	92.89	46.83	87.89	57.55	88.51	57.54
Energy*	92.1	48.28	84.63	72.91	91.88	51.37	88.17	57.95	89.20 (↑0.69)	57.63 (↓0.09)
GEN [11]	94.96	31.93	91.71	46.16	96.77	19.99	89.65	55.55	93.27	38.41
GEN*	93.72	42.54	92.73	40.56	94.14	42.33	88.92	62.9	92.38 (↓0.89)	47.08 (↑8.67)
MCM [20]	95.08	30.31	91.96	43.6	96.69	19.98	89.82	52.7	93.39	36.65
MCM*	94.39	35.87	92.89	41.34	94.54	36.82	89.54	58.45	92.84 (↓0.55)	43.12 (↑6.47)
ViT-B/32										
MSP [6]	91.74	39.28	88.59	55.41	92.85	35.47	85.99	58.95	89.79	47.28
MSP*	92.8	36.6	88.56	59.81	94.35	28.85	86.9	59.2	90.65 (↑0.86)	46.11 (↓1.17)
MaxLogit [7]	92.21	42.11	81.97	77.03	93.46	42.16	87.05	60.15	88.67	55.36
MaxLogit*	93.52	35.0	85.27	70.29	94.86	31.17	87.11	59.85	90.19 (↑1.52)	49.08 (↓6.28)
Energy [8]	90.48	51.19	78.52	82.33	91.59	52.99	85.58	64.75	86.54	62.81
Energy*	91.77	44.4	82.32	76.05	93.08	40.78	85.49	64.0	88.16 (↑1.62)	56.31 (↓6.50)
GEN [11]	93.94	36.21	90.48	51.05	96.05	23.02	88.2	61.05	92.17	42.83
GEN*	93.47	40.46	91.97	44.63	95.84	27.08	87.27	67.0	92.14 (↓0.03)	44.79 (↑1.96)
MCM [20]	94.03	35.02	90.75	51.69	95.91	24.07	88.41	60.0	92.28	42.70
MCM*	93.79	36.67	91.74	47.48	95.91	24.61	87.71	62.05	92.29 (↓0.01)	42.70 (↓0.00)
ViT-L/14										
MSP [6]	94.12	34.34	90.69	50.89	95.52	29.3	90.01	50.2	92.58	41.18
MSP*	95.55	26.41	92.65	43.97	96.31	21.74	91.35	46.3	93.97 (↑1.39)	34.60 (↓6.58)
MaxLogit [7]	93.97	38.39	83.83	75.14	94.95	34.93	90.58	51.95	90.83	50.10
MaxLogit*	94.99	30.29	88.79	58.39	95.79	25.28	91.31	45.6	92.72 (↑1.89)	39.89 (↓10.21)
Energy [8]	92.55	48.74	81.14	80.23	93.5	44.9	89.5	56.4	89.17	57.57
Energy*	93.11	43.28	86.12	66.59	94.17	36.71	89.86	51.85	90.82 (↑1.65)	49.61 (↓7.96)
GEN [11]	95.21	30.75	91.11	49.96	96.47	23.94	90.58	54.65	93.34	39.83
GEN*	95.3	31.46	94.02	37.34	95.81	30.51	90.64	54.7	93.94 (↑0.60)	38.50 (↓1.33)
MCM [20]	95.36	30.58	91.4	50.06	96.6	23.92	90.87	52.75	93.56	39.33
MCM*	95.64	28.22	94.06	38.39	96.2	26.17	91.1	51.45	94.25 (↑0.69)	36.06 (↓3.27)
ResNet-50										
MSP [6]	89.51	51.85	88.26	55.5	91.43	48.57	84.69	62.4	88.47	54.58
MSP*	91.93	39.58	89.22	53.88	92.76	38.11	85.19	62.6	89.78 (↑1.31)	48.54 (↓6.04)
MaxLogit [7]	90.0	55.96	82.22	80.27	91.35	60.71	84.93	64.3	87.12	65.31
MaxLogit*	91.31	49.27	83.16	78.97	91.31	51.58	84.63	65.15	87.60(↑0.48)	61.24 (↓4.07)
Energy [8]	88.03	66.67	78.26	87.07	88.94	74.98	83.26	70.75	84.62	74.87
Energy*	88.78	61.34	78.76	86.94	88.48	66.61	82.68	69.65	84.68 (↑0.06)	71.13 (↓3.74)
GEN	93.41	38.09	91.76	43.91	96.57	21.46	87.02	60.85	92.19	41.08
GEN*	93.01	39.44	92.83	37.38	95.09	29.07	86.4	63.2	91.83 (↓0.36)	42.27 (↑1.19)
MCM [20]	93.11	41.39	91.73	46.18	95.96	28.23	87.07	60.9	91.97	44.17
MCM*	93.18	37.5	92.46	40.68	94.94	29.72	86.61	60.95	91.80 (↓0.17)	42.21 (↓1.96)
ResNet-101										
MSP [6]	92.11	39.97	89.67	50.56	93.85	32.87	86.14	58.5	90.44	45.48
MSP*	92.7	33.48	90.47	45.97	92.63	33.42	86.77	57.35	90.64 (↑0.20)	42.55 (↓2.93)
MaxLogit [7]	91.49	48.24	81.0	82.07	91.85	49.16	85.88	63.0	87.56	60.62
MaxLogit*	91.04	50.64	83.34	76.3	90.05	53.25	85.09	64.0	87.38 (↓0.18)	61.05 (↑0.43)
Energy [8]	88.93	63.41	75.79	89.48	88.71	68.43	83.78	69.95	84.30	72.82
Energy*	87.36	66.96	77.88	86.51	86.29	72.89	82.4	71.75	83.48 (↓0.82)	74.53 (↑1.71)
GEN [11]	94.24	30.52	91.75	41.53	96.0	22.42	87.6	58.55	92.40	38.25
GEN*	93.38	38.03	92.91	36.67	93.83	38.71	86.75	63.0	91.72 (↓0.68)	44.10 (↑5.85)
MCM [20]	94.21	31.79	91.92	42.42	95.93	23.89	87.9	58.2	92.49	39.08
MCM*	93.68	38.16	92.75	41.78	94.06	37.65	87.3	64.0	91.95 (↓1.24)	45.40 (↑6.32)

Table 11: *Per-Dataset Performance of Various OOD Methods and the Ones Enhanced with TAG.* We set $M = 10$ (and $\tau = 0.01$ for MSP, Energy and GEN). The ID dataset is **ImageNet-100** [42]. Green indicates improvement and red indicates degradation.

Models + OOD Method	OpenImage-O		Texture		iNaturalist		ImageNet-O		Average	
	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓
ViT-B/16										
MSP [6]	86.39	51.51	81.57	61.12	87.53	49.66	75.92	75.15	82.85	59.36
MSP*	89.18	45.71	84.13	60.04	89.69	48.09	77.52	77.2	85.13 (↑2.28)	57.76 (↓1.6)
MaxLogit [7]	88.54	55.11	74.94	83.28	90.24	56.8	77.64	76.8	82.84	68.00
MaxLogit*	89.8	50.14	80.33	76.8	89.29	60.1	78.49	76.65	84.48 (↑1.64)	65.92 (↓2.08)
Energy [8]	85.45	69.16	69.22	90.66	87.03	75.4	75.35	81.15	79.26	79.09
Energy*	85.75	70.17	74.53	87.42	84.83	79.56	75.81	82.55	80.23 (↑0.97)	79.92 (↑0.83)
GEN [11]	92.19	39.88	86.95	55.37	94.02	33.23	81.64	71.9	88.70	50.09
GEN*	91.61	41.7	86.56	59.65	92.3	43.1	80.86	75.35	87.83 (↓0.87)	54.95 (↑4.86)
MCM [20]	91.65	43.06	87.09	56.05	94.39	32.6	79.61	75.9	88.19	51.90
MCM*	90.78	47.01	88.79	53.28	91.78	47.73	79.53	79.75	87.72 (↓0.47)	56.94 (↑5.04)
ViT-B/32										
MSP [6]	83.89	58.71	79.12	67.05	85.57	53.92	70.6	80.3	79.79	65.00
MSP*	86.3	56.09	81.47	65.91	87.4	54.67	72.94	81.85	82.03 (↑2.24)	64.63 (↓0.37)
MaxLogit [7]	86.22	59.58	72.35	85.27	88.14	63.45	73.4	81.1	80.03	72.35
MaxLogit*	88.5	53.19	77.32	80.27	89.98	57.67	74.35	80.8	82.54 (↑2.51)	67.98 (↓4.37)
Energy [8]	83.06	71.88	66.93	92.27	84.44	78.99	71.51	85.00	76.48	82.03
Energy*	85.15	66.96	71.64	89.32	86.25	73.01	71.88	84.65	78.73 (↑2.25)	78.48 (↓3.55)
GEN [11]	90.42	46.55	85.48	61.18	92.6	39.17	78.05	78.45	86.64	56.34
GEN*	89.47	51.58	84.23	67.19	91.31	49.72	77.54	82.1	85.64 (↓1.00)	62.65 (↑6.31)
MCM [20]	90.1	47.22	85.77	59.59	93.51	34.4	75.85	80.6	86.31	55.45
MCM*	89.45	51.69	87.39	56.61	92.25	44.95	76.15	83.05	86.31 (↑0.00)	59.08 (↑3.63)
ViT-L/14										
MSP [6]	89.85	41.76	83.13	59.11	91.52	37.14	80.74	65.65	86.31	50.92
MSP*	92.42	34.34	85.92	55.0	93.61	31.4	82.84	66.55	88.70 (↑2.39)	46.82 (↓4.10)
MaxLogit [7]	90.27	50.82	74.63	83.41	91.66	49.91	81.08	71.5	84.41	63.91
MaxLogit*	91.54	44.17	80.05	74.38	92.57	43.91	81.44	70.8	86.40 (↑1.99)	58.32 (↓5.59)
Energy [8]	87.31	67.56	69.63	89.63	88.48	68.57	78.89	76.8	81.08	75.64
Energy*	87.64	65.17	74.85	83.84	88.72	65.56	78.63	77.7	82.46 (↑1.38)	73.07 (↓2.57)
GEN [11]	93.96	29.97	87.48	53.59	95.76	22.77	84.75	62.95	90.49	42.32
GEN*	93.72	31.68	86.85	55.85	94.79	28.37	84.33	67.35	89.92 (↓0.57)	45.81 (↑3.49)
MCM [20]	93.08	35.04	86.62	55.66	94.96	28.3	82.59	68.55	89.31	46.89
MCM*	93.05	36.96	88.55	52.02	93.9	37.22	82.04	73.8	89.39 (↑0.08)	50.00 (↑3.11)
ResNet-50										
MSP [6]	82.58	62.28	79.87	65.6	85.99	55.16	68.42	86.6	79.22	67.41
MSP*	84.7	58.86	81.79	62.81	87.82	53.16	70.1	86.5	81.10 (↑1.88)	65.33 (↓2.08)
MaxLogit [7]	84.24	68.76	72.0	89.71	86.0	76.01	71.12	85.95	78.34	80.11
MaxLogit*	85.18	67.38	75.1	88.64	84.89	77.37	71.19	86.2	79.09 (↑0.75)	79.90 (↓0.21)
Energy [8]	80.59	81.2	65.51	94.46	81.02	90.8	69.3	88.2	74.11	88.66
Energy*	81.0	80.52	67.98	95.45	78.74	91.04	68.91	87.9	74.16 (↑0.05)	88.73 (↑0.07)
GEN [11]	88.98	52.63	86.45	58.86	92.35	42.02	76.31	83.25	86.02	59.19
GEN*	87.81	56.86	84.48	66.32	90.74	51.88	74.81	87.05	84.46 (↓1.56)	65.53 (↑6.34)
MCM [20]	89.18	50.91	86.97	58.51	93.75	34.61	74.46	84.65	86.09	57.17
MCM*	88.09	56.48	88.17	55.43	91.43	50.11	74.45	86.05	85.53 (↓0.56)	62.02 (↑4.85)
ResNet-101										
MSP [6]	83.53	60.68	79.36	66.94	82.35	61.86	70.47	82.4	78.93	67.97
MSP*	85.39	59.03	82.62	61.24	85.61	58.88	71.72	84.4	81.34 (↑2.41)	65.89 (↓2.08)
MaxLogit [7]	83.94	72.86	69.61	91.96	82.33	82.9	71.78	86.05	76.91	83.44
MaxLogit*	84.69	72.62	75.47	88.53	83.24	79.85	72.08	86.7	78.87 (↑1.96)	81.92 (↓1.52)
Energy [8]	79.56	85.26	62.19	97.23	77.53	94.16	69.36	87.75	72.16	91.10
Energy*	79.2	84.13	67.19	95.27	77.11	92.32	69.15	89.35	73.16 (↑1.00)	90.27 (↓0.83)
GEN [11]	89.24	52.86	84.99	62.46	89.58	53.15	77.23	82.25	85.26	62.68
GEN*	88.48	55.89	85.01	65.33	89.12	57.53	76.31	84.15	84.73 (↓0.53)	65.72 (↑3.04)
MCM [20]	88.82	54.82	86.26	59.28	89.93	53.35	75.15	83.6	85.04	62.76
MCM*	88.38	56.27	88.25	51.53	89.21	57.12	75.36	84.3	85.30 (↑0.26)	62.31 (↓0.45)

Table 12: *Per-Dataset Performance of Various OOD Methods and the Ones Enhanced with TAG.* We set $M = 10$ (and $\tau = 0.01$ for MSP, Energy and GEN). The ID dataset is **ImageNet-1k** [42]. Green indicates improvement and red indicates degradation.

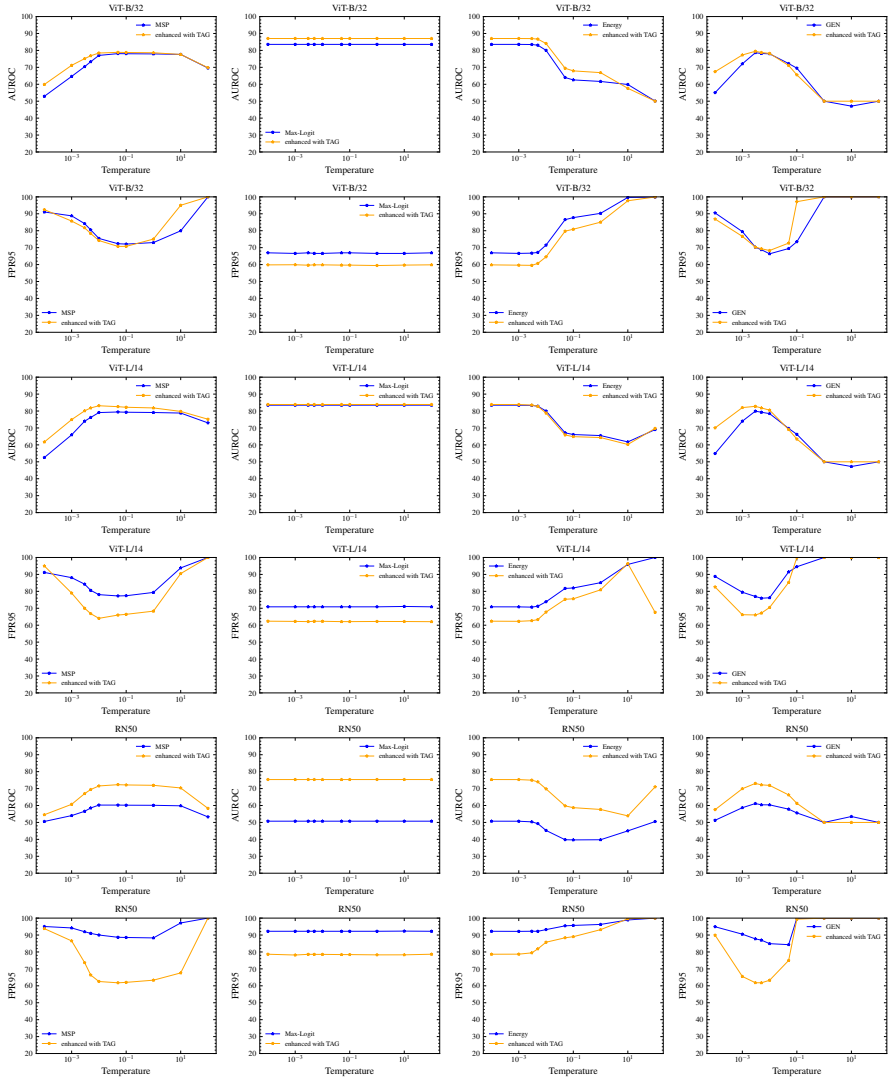

Figure 7: *Averaged Performance (over 5 OOD Datasets) of TAG Applied with Different Temperature τ.* The ID dataset is **CIFAR-100**. The evaluated models are **ViT-B/32**, **ViT-L/14**, and **ResNet-50.** Every two rows represent the performance from the same model in terms of AUROC and FPR95, respectively. Each column denotes different score functions including MSP [6], MaxLogit [7], Energy [8], and GEN [11] (from left to right).

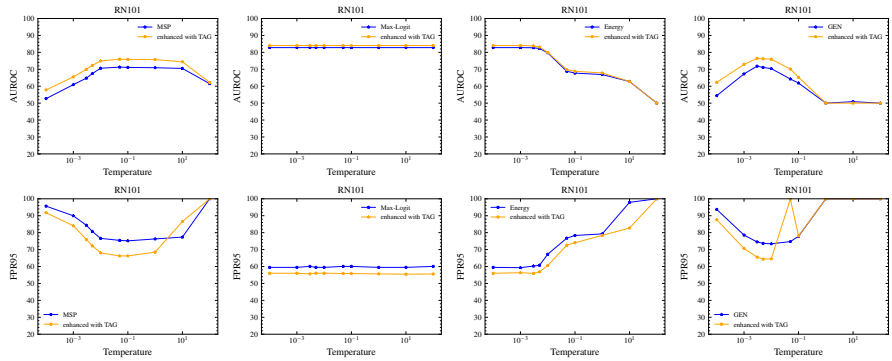

Figure 8: *Averaged Performance (over 5 OOD Datasets) of TAG Applied with Different Temperature τ.* The ID dataset is **CIFAR-100** and the model is **ResNet-101**. TAG performance in terms of AUROC values (top row) and FPR95 (bottom row). Each column denotes different score functions including MSP [6], MaxLogit [7], Energy [8], and GEN [11] (from left to right).

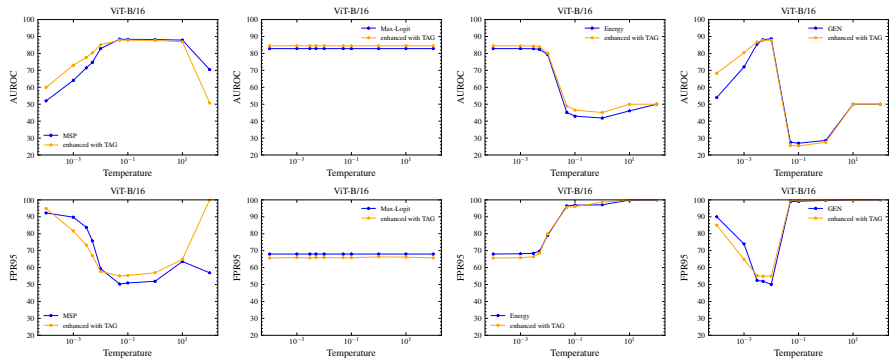

Figure 9: *Averaged Performance (over 4 OOD Datasets) of TAG Applied with Different Temperature τ.* The ID dataset is **ImageNet-1k** and the model is **ViT-B/16**.TAG performance in terms of AUROC values (top row) and FPR95 (bottom row). Each column denotes different score functions including MSP [6], MaxLogit [7], Energy [8], and GEN [11] (from left to right).

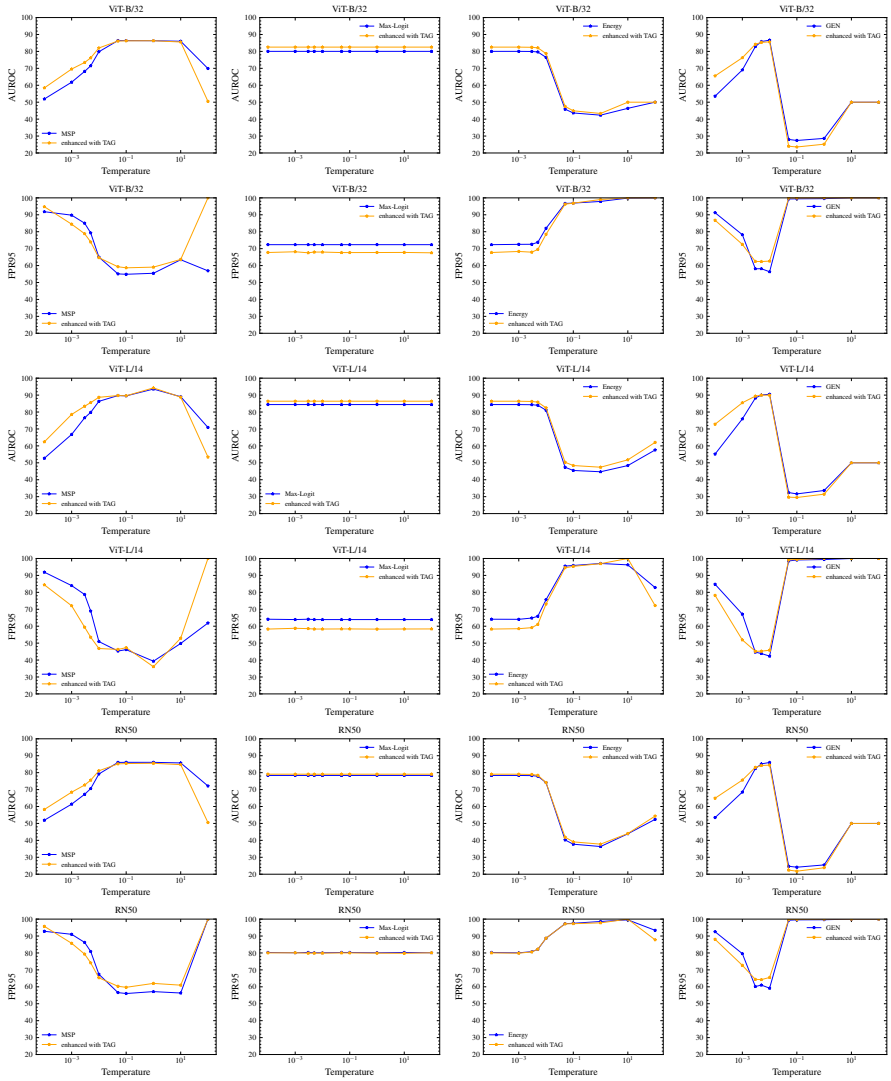

Figure 10: *Averaged Performance (over 4 OOD Datasets) of TAG Applied with Different Temperature τ.* The ID dataset is **ImageNet-1k**. The evaluated models are **ViT-B/32**, **ViT-L/14**, and **ResNet-50**. Every two rows represent the performance from the same model in terms of AUROC and FPR95, respectively. Each column denotes different score functions including MSP [6], MaxLogit [7], Energy [8], and GEN [11] (from left to right).

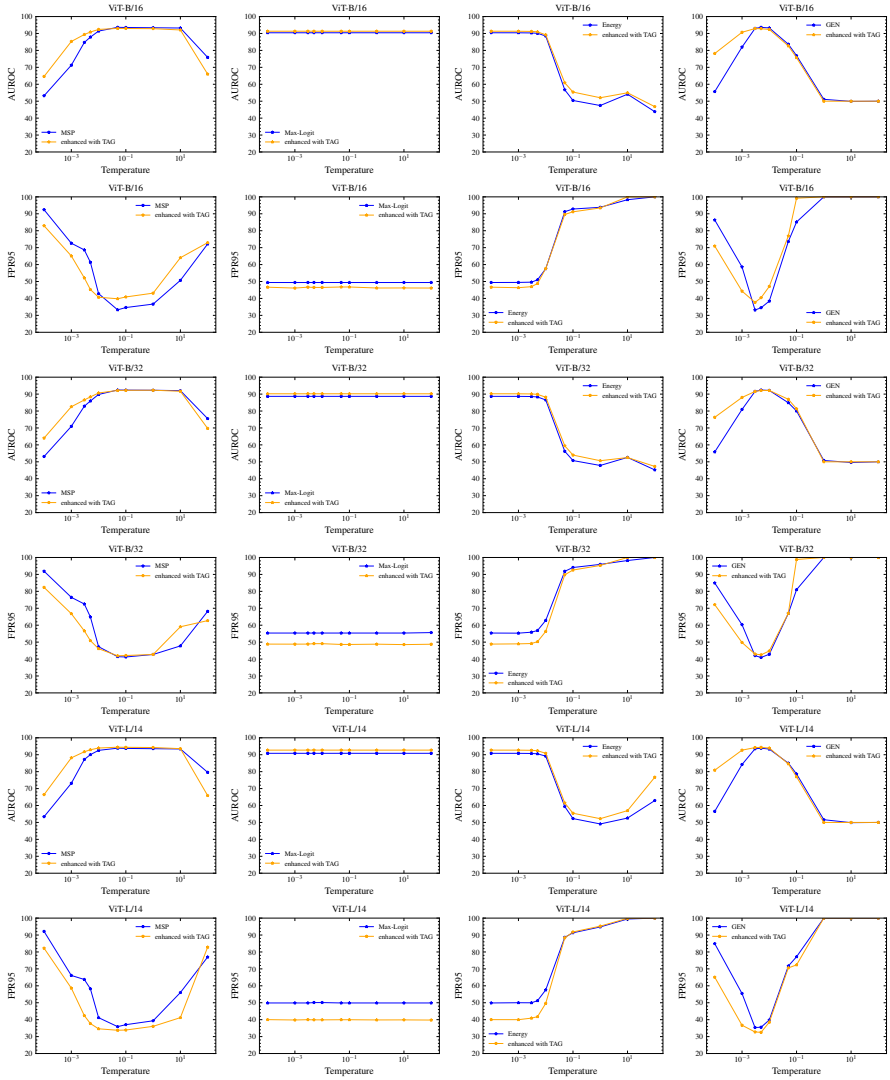

Figure 11: *Averaged Performance (over 4 OOD Datasets) of TAG Applied with Different Temperature τ.* The ID dataset is **ImageNet-100**. The evaluated models are **ViT-B/16, ViT-B/32, and ViT-L/14.** Every two rows represent the performance from the same model in terms of AUROC and FPR95, respectively. Each column denotes different score functions including MSP [6], MaxLogit [7], Energy [8], and GEN [11] (from left to right).

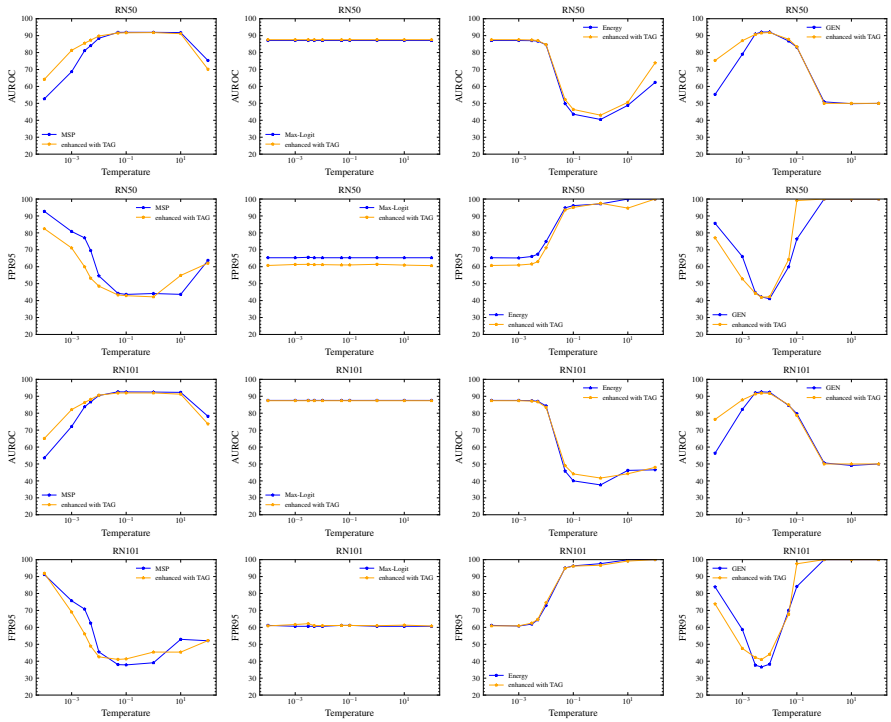

Figure 12: *Averaged Performance (over 4 OOD Datasets) of TAG Applied with Different Temperature τ.* The ID dataset is **ImageNet-100** and the model is **ResNet-50** and **ResNet-101**. Every two rows represent the performance from the same model in terms of AUROC and FPR95, respectively. Each column denotes different score functions including MSP [6], MaxLogit [7], Energy [8], and GEN [11] (from left to right).

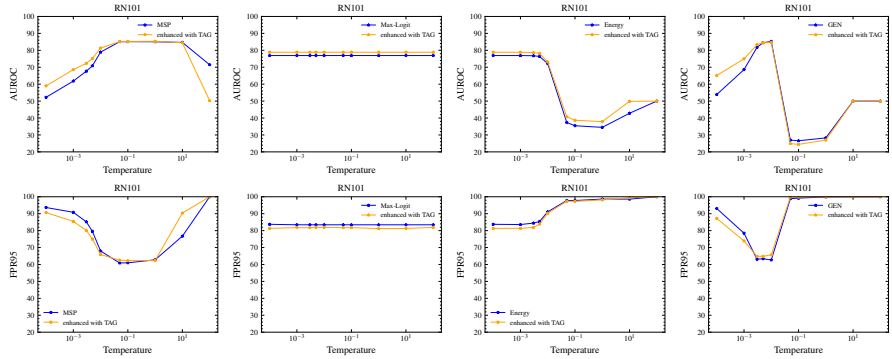

Figure 13: *Averaged Performance (over 4 OOD Datasets) of TAG Applied with Different Temperature τ.* The ID dataset is **ImageNet-1k** and the model is **ResNet-101**. TAG performance in terms of AUROC values (top row) and FPR95 (bottom row). Each column denotes different score functions including MSP [6], MaxLogit [7], Energy [8], and GEN [11] (from left to right).

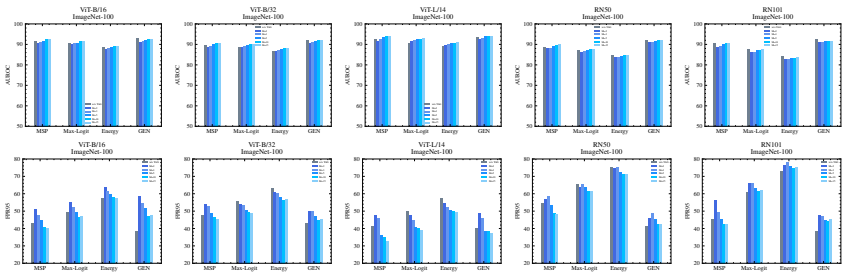

Figure 14: *Averaged Performance (over 4 OOD datasets) of TAG Varying with Different Augmentations M across Different Architectures.* The ID dataset is **ImageNet-100**. (top row) AUROC values and (bottom row) FPR95 values.

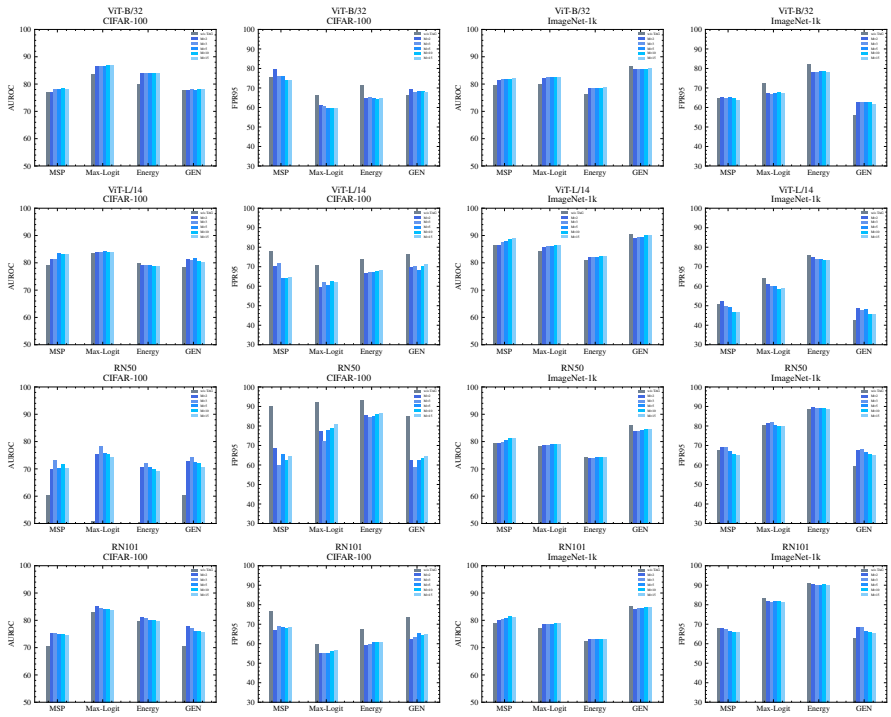

Figure 15: *Averaged Performance of TAG Varying with Different Augmentations M across Different Architectures.* The left two column corresponds to CIFAR-100 [41] dataset, and the right two columns corresponds to ImageNet-1k [42]. The averages are computed across 5 OOD datasets (CIFAR-100) and 4 datasets (ImageNet-1k), respectively.

References

[1] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[2] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *Information Processing in Medical Imaging (IPMI)*, 2017.

[3] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan, "Do deep generative models know what they don't know?" In *International Conference on Learning Representations (ICLR)*, 2019.

[4] J. Yang, K. Zhou, Y. Li, and Z. Liu, "Generalized out-of-distribution detection: A survey," *arXiv:2110.11334*, 2021.

[5] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *Advances in Neural Information Processing Systems*, 2018.

[6] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[7] D. Hendrycks, S. Basart, M. Mazeika, *et al.*, "Scaling out-of-distribution detection for real-world settings," in *International Conference on Machine Learning (ICML)*, 2022.

[8] W. Liu, X. Wang, J. Owens, and Y. Li, "Energy-based out-of-distribution detection," in *Advances in Neural Information Processing Systems*, 2020.

[9] R. Huang, A. Geng, and Y. Li, "On the importance of gradients for detecting distributional shifts in the wild," in *Advances in Neural Information Processing Systems*, 2021.

[10] H. Wang, Z. Li, L. Feng, and W. Zhang, "Vim: Out-of-distribution with virtual-logit matching," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[11] X. Liu, Y. Lochman, and C. Zach, "Gen: Pushing the limits of softmax-based out-of-distribution detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[12] Y. Song, N. Sebe, and W. Wang, "Rankfeat: Rank-1 feature removal for out-of-distribution detection," in *Advances in Neural Information Processing Systems*, 2022.

[13] Y. Sun, C. Guo, and Y. Li, "React: Out-of-distribution detection with rectified activations," in *Advances in Neural Information Processing Systems*, 2021.

[14] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," in *International Conference on Learning Representations (ICLR)*, 2019.

[15] X. Du, Y. Sun, X. Zhu, and Y. Li, "Dream the impossible: Outlier imagination with diffusion models," in *Advances in Neural Information Processing Systems*, 2023.

[16] M. Cook, A. Zare, and P. Gader, "Outlier detection through null space analysis of neural networks," *arXiv:2007.01263*, 2020.

[17] A. Radford, J. W. Kim, C. Hallacy, *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning (ICML)*, 2021.

[18] S. Fort, J. Ren, and B. Lakshminarayanan, "Exploring the limits of out-of-distribution detection," in *NeurIPS*, 2021.

[19] S. Esmaeilpourcharandabi, B. Liu, E. Robertson, and L. Shu, "Zero-shot open set detection by extending clip," in *AAAI*, 2021.

[20] Y. Ming, Z. Cai, J. Gu, Y. Sun, W. Li, and Y. Li, "Delving into out-of-distribution detection with vision-language representations," in *Advances in Neural Information Processing Systems*, 2022.

[21] A. Miyai, Q. Yu, G. Irie, and K. Aizawa, "Zero-shot in-distribution detection in multi-object settings using vision-language foundation models," in *AAAI*, 2023.

[22] Y. Dai, H. Lang, K. Zeng, F. Huang, and Y. Li, "Exploring large language models for multi-modal out-of-distribution detection," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

[23] K. Bibas, M. Feder, and T. Hassner, "Single layer predictive normalized maximum likelihood for out-of-distribution detection," in *Advances in Neural Information Processing Systems*, 2021.

[24] Y. Sun, Y. Ming, X. Zhu, and Y. Li, "Out-of-distribution detection with deep nearest neighbors," in *International Conference on Machine Learning (ICML)*, 2022.

[25] G. Xia and C.-S. Bouganis, "Augmenting softmax information for selective classification with out-of-distribution data," in *Asian Conference on Computer Vision (ACCV)*, 2022.

[26] Y.-C. Hsu, Y. Shen, H. Jin, and Z. Kira, "Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[27] Y. Ming, Y. Sun, O. Dia, and Y. Li, "Cider: Exploiting hyperspherical embeddings for out-of-distribution detection," *arXiv:2203.04450*, 2022.

[28] Y. Ming, Y. Fan, and Y. Li, "Poem: Out-of-distribution detection with posterior sampling," in *International Conference on Machine Learning*, 2022.

[29] X. Du, Z. Wang, M. Cai, and S. Li, "Vos: Learning what you don't know by virtual outlier synthesis," in *International Conference on Learning Representations (ICLR)*, 2022.

[30] A. Djurisic, N. Bozanic, A. Ashok, and R. Liu, "Extremely simple activation shaping for out-of-distribution detection," in *International Conference on Learning Representations (ICLR)*, 2023.

[31] K. Xu, R. Chen, G. Franchi, and A. Yao, "Scaling for training time and post-hoc out-of-distribution detection enhancement," in *International Conference on Learning Representations (ICLR)*, 2024.

[32] A.-A. Papadopoulos, M. R. Rajati, N. Shaikh, and J. Wang, "Outlier exposure with confidence control for out-of-distribution detection," *Neurocomputing*, 2021.

[33] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *International Conference on Learning Representations (ICLR)*, 2018.

[34] Q. Wang, J. Ye, F. Liu, *et al.*, "Out-of-distribution detection with implicit outlier transformation," in *International Conference on Learning Representations (ICLR)*, 2023.

[35] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[36] H. Wang, Y. Li, H. Yao, and X. Li, "Clipn for zero-shot ood detection: Teaching clip to say no," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[37] S. Menon and C. Vondrick, "Visual classification via description from large language models," *International Conference on Learning Representations (ICLR)*, 2023.

[38] K. Roth, J. M. Kim, A. S. Koepke, O. Vinyals, C. Schmid, and Z. Akata, "Waffling around for performance: Visual classification with random words and broad concepts," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[39] Y. Bang, S. Cahyawijaya, N. Lee, *et al.*, "A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity," in *International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023.

[40] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," in *Advances in Neural Information Processing Systems*, 2019.

[41] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[42] O. Russakovsky, J. Deng, H. Su, *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, 2015.

[43] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Ng, "Reading digits in natural images with unsupervised feature learning," in *Advances in Neural Information Processing Systems*, 2011.

[44] P. Xu, K. A. Ehinger, Y. Zhang, A. Finkelstein, S. R. Kulkarni, and J. Xiao, "Turkergaze: Crowdsourcing saliency with webcam based eye tracking," *arXiv:1504.06755*, 2015.

[45] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[46] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[47] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv:1506.03365*, 2016.

[48] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, "Natural adversarial examples," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[49] I. Krasin, T. Duerig, N. Alldrin, *et al.*, "Openimages: A public dataset for large-scale multi-label and multi-class image classification.," *Dataset available from https://github.com/openimages*, 2017.

[50] G. Van Horn, O. Mac Aodha, Y. Song, *et al.*, "The inaturalist species classification and detection dataset," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[51] S. Yun, S. J. Oh, B. Heo, D. Han, J. Choe, and S. Chun, "Re-labeling imagenet: From single to multi-labels, from global to localized labels," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[52] C. Oh, J. So, H. Byun, *et al.*, "Geodesic multi-modal mixup for robust fine-tuning," in *Advances in Neural Information Processing Systems*, 2023.

[53] J. Zhang, J. Yang, P. Wang, *et al.*, "Openood v1.5: Enhanced benchmark for out-of-distribution detection," in *Advances in Neural Information Processing Systems Workshop on Distribution Shifts*, 2023.

PAPER F

Energy-Guided Decoding for Object Hallucination Mitigation

Xixi Liu, Ailin Deng, Christopher Zach

Submitted for Review, 2025

The layout has been revised.

Abstract

To ensure the reliable deployment of large vision language models (LVLMs) in the real world, particularly for safety-critical applications, it is essential to resolve the issue of hallucination, *i.e.*, LVLMs occasionally generating contents that are not grounded in the visual inputs. Existing methods either demand sophisticated modifications to visual inputs [1], are restricted to specific decoding strategies [2], or rely on knowledge from other models [3]. In this work, we identify a significant imbalance in the yes ratio, *i.e.*, the fraction of "yes" answers among the total number of questions, within VLMs. In order to mitigate this hallucinatory behavior we propose an energy-based decoding method, which dynamically select the hidden states from the layer with minimal energy score. It is simple and effective in reducing the bias for the yes ratio and boosting performance across three discriminative benchmarks (POPE, MME, and MMVP). Our method consistently improves accuracy and F1 score on POPE benchmark across two commonly used VLMs over three baseline methods. The average accuracy improvement is 4.37% compared to the greedy decoding. Moreover, the proposed method is less biased in terms of yes ratio as shown in Figure 1.

1 Introduction

Large language models (LLMs) such as ChatGPT [5] have shown great capability spanning over a wide range of domains including but not limited to search and personalized recommendation, virtual assistants, fraud detection, and coding assistance tools. Meanwhile, vision-language models (VLMs) such as GPT-4V(ision) can describe the real world e.g. to visually impaired people **gpt4v**, [6]–[8]. However, all those models, also known as foundation models, suffer from the issue of hallucination. Hallucination in LLMs refers to the problem that either the output of LLMs is inconsistent with the source content in context, or the LLMs generate a response that is not grounded by the pre-training dataset [9]. Not surprisingly, all VLMs are also affected by

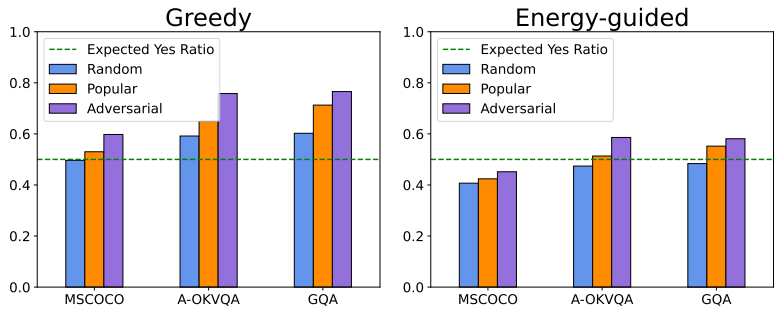

Figure 1: *"Yes" Ratio Comparison* using greedy decoding (left) and energy-guided decoding (right) across three datasets over three settings including *random*, *popular*, and *adversarial*. LLaVA-1.5 [4] is employed as the VLM backbone. The optimal "yes" ratio is 50% (green dashed line).

hallucination. Here it refers to the scenarios that VLMs occasionally generate responses that are not supported by the visual input. A recent survey [10] categories the hallucinations in VLMs, in particular, object-related hallucinations into the following groups: 1) category, where the VLM identifies incorrect or non-existing objects in the image; 2) attribute, where wrong description such as color and shape for the given visual input are generated; 3) relation, where incorrect relationship or interactions between objects are reported. The existing benchmarks used to assess the extent of hallucination in VLMs can be roughly categorized into discriminative tasks including POPE [11], MME [12] and MMVP [13], and generative tasks such as GPT4-Assisted Visual Instruction Evaluation (GAVIE) [14]. In the work, we focus on the discriminative tasks, and the corresponding dataset summary is shown in Table 2. Considering that most benchmarks primarily address object hallucinations at the category level, with limited coverage of the other levels [12], [13]. Therefore, in this work, we focus on mitigating hallucinations mainly at the category level.

The problem of object hallucination mitigation can be traced back to [18], which is the initial work to investigate the issue of object hallucination in the image captioning task. The cause of hallucination in VLMs is more complex. First, the hallucination might be induced by the language prior, which is analogous to hallucinations in LLMs [15]. Second, a number of possible causes are

Table 1: *Comparison of Hallucination Mitigation Methods.* Compared with previous method, our method requires minimal effort to mitigate object hallucination.

Method	Free of					
	pre-defined layers	visual editing	prompt tuning	specific decoding	external knowledge	contrastive decoding
DoLa [15]	✗	✓	✓	✓	✓	✗
ICD [16]	✓	✓	✗	✓	✓	✗
CGD [3]	✓	✓	✓	✓	✗	✓
VCD [1]	✓	✗	✓	✓	✓	✗
OPERA [2]	✓	✓	✓	✗	✓	✓
HALC [17]	✓	✓	✓	✓	✗	✗
Energy-guided (Ours)	✓	✓	✓	✓	✓	✓

related to the utilized visual encoders, such as its capacity [13], the quality of vision-language instruction-following data [19], and the training objectives employed for feature alignment. Compared to prior works, our method does not require contrastive decoding [1], [2], [16], [17], specific decoding strategies [2], [3], corrupted images [1], or prompt engineering [16]. It is highly efficient, which only requires one single forward pass to calculate the energy score at each layer. The hidden states from the layer with minimal energy score are then utilized for subsequent decoding.

Contributions

1. We empirically observe the inherent bias in terms of yes ratio that exists in the language decoder, particularly, for out-of-distribution datasets including Q-OKVQA [20] and GQA [21], cf. Fig. 1(left).

2. Further, we propose a simple and effective decoding strategy termed as energy-guided decoding for mitigating object hallucination, mainly at the category level. It does not require fine-tuning, contrastive decoding, or external models. Yes it performs very well resulting in a less biased yes ratio, cf. Fig. 1 (right) and improved accuracy and F1 score, cf. Fig. 2 (right).

2 Related work

Contrastive decoding in VLMs Contrastive decoding was initially proposed to mitigate hallucination in LLMs [22]. Specifically, it leverages two LLMs

with different capabilities (*i.e.*, one is the "expert" and the other is "amateur"). By contrasting the predictive distribution from two LLMs, the token that captures the largest difference is selected for generation. Similarly, DoLa [15] follows the same principle yet without external knowledge from other LLMs. DoLa [15] observes that the knowledge bias mainly comes from early layers and then utilize this phenomenon to mitigate hallucination by contrasting the predictive distributions induced by different layers within one LLM. Naturally, a similar principle can also be applied to VLMs [1], [16], [17], [23]. VCD [1] observes that perturbed images (e.g. obtained by adding Gaussian noise to the original ones) have an increased tendency to hallucinate (*i.e.*, the winning logits generated from the perturbed image are more often induced by a language prior). Therefore, the final logits are a linear combination of the ones induced by the original image and perturbed image, respectively. VDD [23] adopts the same principle as VCD but with an additional calibration step. To be specific, a weight matrix W is learned to transform the predictive distribution produced from the case of replacing the noisy image with a dummy test with no images to be a uniform distribution for each answer. Afterwards, the same criterion as VCD is applied, *i.e.*, the final logit is a linear combination of the calibrated logits with and without the original image. Instruction contrastive decoding (ICD) [16] extends the contrastive principle to the introductions/prompts literally by adding a prefix (e.g., `You are a confused object detector`) to the standard prompt to further amplify the hallucination. Similarly, the calculation of the final logit is the same as VCD [1] and VDD [23]. Most contrastive decoding methods for hallucination mitigation operate within internal states and require a contrasted distribution from either a distorted visual input [1], [23], or a pre-defined layer bucket [15], or prompt engineering [16].

Non-contrastive decoding in VLMs Another line of hallucination mitigation methods does not rely on contrasting another logit distribution [2], [3]. CGD [3] aims to mitigate object hallucination on a sentence level. Particularly, it leverages the powerful vision-language alignment capabilities of CLIP to identify sentences that are better aligned with the corresponding visual embeddings. This ensures that the generated responses not only have higher sentence likelihood but also higher CLIP scores. Therefore, the generated sentences are less hallucinated. However, its performance gain highly relies on

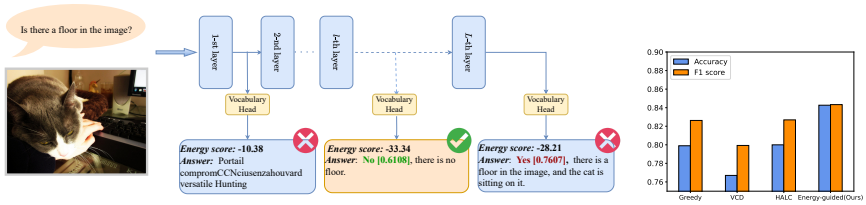

Figure 2: *An illustration of Energy-guided Decoding.* We observe that the hidden states from the layer with minimal energy score generates more accurate response. We also report the confidence of " yes" and " no" measuring by the corresponding token probability. The right barplot shows the overall performance comparison in terms of accuracy and F1 score on GQA dataset with *adversarial* setting and the VLM backbone is LLaVA-1.5 [4].

the capability of external models. Further, the possible decoding methods are redistricted to nucleus sampling [24] and beam search in order to create the candidate sentences. OPERA [2] observes the phenomenon that the presence of hallucination correlates with certain "knowledge aggregation patterns", *i.e.*, VLMs tend to generate new tokens by focusing on a few summary tokens but not necessarily taking all the previous tokens into account. Therefore, the hallucination is mitigated by penalizing the "over-trust" logit. However, the hysteresis of beam-search necessitates a mechanism named retrospection-allocation, i.e., the decoding procedure may roll back to the identified summary token and select other candidates for the next token prediction except for the candidates selected before. Consequently, OPERA [2] iteratively operates with the beam-search decoding, which results in high-computational demand at the inference stage but also severely restricts its applicable scenarios. Our method is highly efficient, which only requires one single forward pass to calculate the energy score at each layer.

Latent representations in language models Understanding the decoding mechanism of transformer-based language decoders has been studied from various perspectives including but not limited to attention maps/patterns [25]–[28] and the intermediate representation [29]–[34] with the application of early exiting [32], [33] or model knowledge editing [35], [36]. Model knowledge editing refers to identifying and removing a (linear) concept subspace from the

representation, preventing any (linear) predictor from recovering the concept. Meanwhile, early exiting in the context LLMs refers to projecting the hidden states extracted at each layer to the learned "unembedding" matrix of the language decoder. By doing this, one can obtain the multiple logit distributions for the following decoding.

Unlike existing hallucination mitigation methods such as VCD [1] (which necessitates generating a sophisticated noisy version of the original visual inputs), OPERA [2] (which relies on the beam-searching decoding mechanism), HALC [17] (requiring a pre-defined layer bucket and an external detector), and MMVP [13] (relies on additional fine-tuning), our method is derived through the lens of internal states of a language decoder. Termed energy-guided decoding, it avoids the need of visual distortion, or prompt engineering, or external detectors making it free from contrastive decoding. More importantly, the energy score at each layer can be computed with a single forward pass, making our method significantly less computationally demanding compared to OPERA [2] and HALC [17] .

3 Methods

3.1 Vision-Language Model Summary

Generally, the input tokens processed by VLMs consist of visual and text tokens. The visual tokens of the input image is denoted by $\{I_1, I_2, I_3, \cdots, I_N\}$ and the corresponding language tokens is denoted by $\{W_1, W_2, W_3, \cdots, W_M\}$. N and M are the corresponding length of visual tokens and language tokens, respectively. Afterwards, the visual tokens and language tokens are concatenated together, which is denoted with $\mathbf{x}$ and regarded as the final input tokens with the length of $T = N + M$. VLMs are commonly trained in an autoregressive manner with a causal attention mask meaning that the prediction of the current token x_t only depends on the previous tokens, formally,

$$\mathbf{h} = \text{VLM}(\mathbf{x}) = \{h_0, h_1, \cdots, h_{T-1}\}, \tag{F.1}$$

where $\mathbf{h}$ is the output state of the final layer of LLM decoder, and the size of h_t is f_{dim}. A learned vocabulary head $\mathcal{H}$ with the size of V_{size} is utilized to obtain the logits. The learned vocabulary head $\mathcal{H}$ plays a similar role as the

penultimate layer of standard discriminative classifier, formally,

$$p(x_t|x_{<t}) = \text{Softmax}[\mathcal{H}(h_t)], \tag{F.2}$$

where $x_{<t}$ denotes the sequence of tokens before t-th position $\{x_i\}_{i=0}^{t-1}$ and $\mathcal{H} \in \mathbb{R}^{f_{\text{dim}} \times V_{\text{size}}}$.

3.2 Empirical Yes Ratio Transfer

The source of hallucination appeared in VLMs can be attributed to (i) the embedded knowledge in the language decoder's parameters (i.e., the cause of hallucination in LLMs [15]); (ii) a limited capacity of the visual encoder [13]; (iii) the quality of vision-language instruction-following data [19]; (iv) the training objectives [37]; and (v) the connector that accounts for the feature alignment [4]. In this work, we focus on the language decoder in the context of VLMs from the perspective of the yes ratio.

Yes ratio transfer We start with the evaluation covering two scenarios—one that includes visual input and one without visual input.[1] Fig. 3 provides some interesting empirical observations:

1. The "yes" ratio—answers labeled "yes" out of total questions—generally increases as tasks grow more challenging, from random to popular to adversarial settings.

2. The "yes" ratio is initially high without visual input, suggesting models are biased toward "yes" due to language priors in this dataset.

These findings indicate that language models' "yes" bias can transfer to VLMs, especially under more challenging, hallucinatory tasks.

Yes/No confidence We further visualize the confidence of "yes" and "no" measuring by the corresponding token probability. One can see from Fig. 4(top row) that the model is overconfident to say "yes" (blue area) compared to say "no" (orange area) across three settings of POPE-GQA [21]. Ideally, the confidence levels for both response alternatives are balanced, such as shown in Fig. 4(bottom row) (which is actually obtained by our proposed method).

[1] The codebase is `https://github.com/haotian-liu/LLaVA/tree/main/llava/eval`

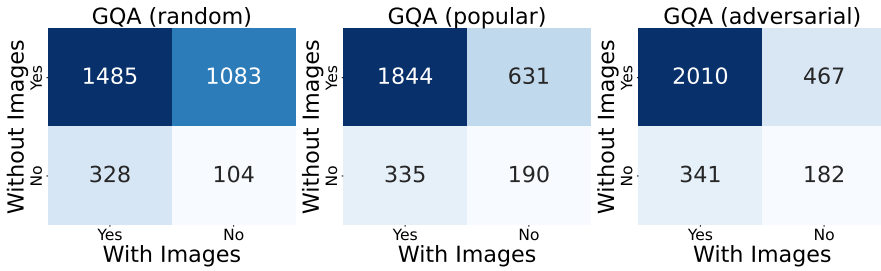

Figure 3: *Transfer of "yes" ratio* from non-visual input to visual inputs using greedy decoding. Three settings of POPE-GQA [11] are utilized including random (left column), popular (middle column), and adversarial (right column).

3.3 Proposed Method

The architectures of transformer-based language decoder enable the feasibility of directly decoding hidden states from each layer into vocabulary space using the model's pre-trained "unembedding" matrix $\mathcal{H}$. This early exiting technique is termed as "logit lens" [32] and has been utilized to include but not limited to analyzing decoding mechanism of transformer-based language decoders [29]–[31] or improving factuality of LLMs [15]. It is empirically shown in [32] that the hidden states from internal layers may already be interpretable. Moreover, DoLa [15] empirically shows the information decoded from earlier layer might be biased to the language prior. We take similar inspiration by considering the learned vocabulary head (also known as unembedding matrix) as the classifier in the standard deep models, and re-interpret the result before Softmax as the logit distribution. In this way, we obtain a logit distribution for each layer.

Energy score We propose to identify the most suitable layer in the language decoder as the layer with the minimal value for its energy score, which is defined as the negated "soft-maximum" of the logits,

$$\begin{aligned} E_\theta(\mathbf{x}) &= -\operatorname{LogSumExp}_y(f_\theta(\mathbf{x})[y]) \\ &:= -\log \sum\nolimits_y \exp(f_\theta(\mathbf{x})[y]), \end{aligned} \tag{F.3}$$

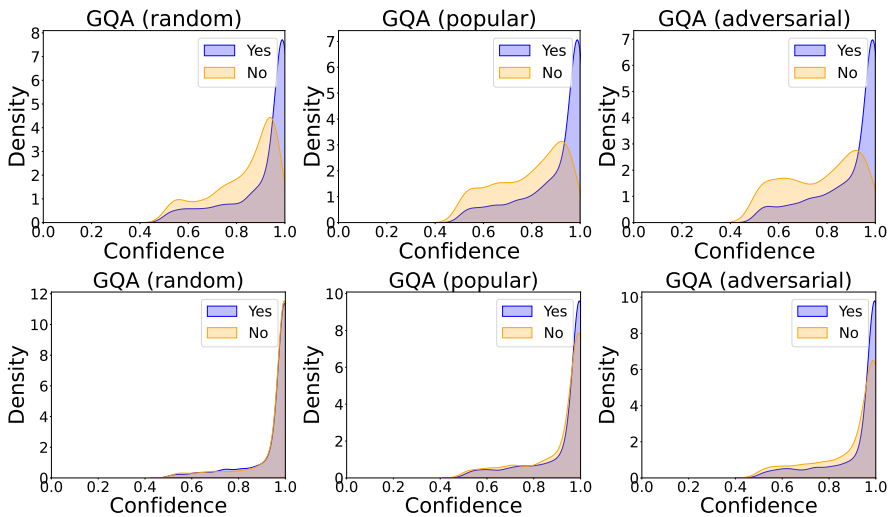

Figure 4: *Kernel Density Estimation* for answers with yes (blue) and the ones with no (orange), using hidden states from the last layer for decoding (top row) and the ones with minimum energy for decoding (bottom row), respectively. Three settings of POPE-GQA [11] are utilized including random (left column), popular (middle column), and adversarial (right column).

for a network f_θ. In our setting the logits are obtained by applying the vocabulary head $\mathcal{H}$ to the feature representation at any given layer. The energy score has successfully been applied for out-of-distribution (OOD) detection [38], in particular to identify samples that are affected by a semantic shift. If a discriminative network is trained appropriately, then the energy score directly corresponds to the negative log-evidence $\log p(\mathbf{x})$ [39]. Alternatively, in certain scenarios, the energy score obtained from solely discriminatively trained networks may correspond to some form of feature log-likelihood [40].

The use of a quantity such as the energy score in this work is motivated by the observations illustrated in Fig. 4: negative ("no") responses are consistently under-confident and a source of the prevalence of "yes" answers in Fig. 3. In order to balance this asymmetry in confidence we therefore neutralize differences in the logit vectors by always using the most confident layer (in terms of the energy score) for the subsequent decoding step.

Energy-guided decoding In the context of VLMs, the input $\mathbf{x}$ includes both the visual input tokens and the paired text tokens. This allows us to compute an energy score for the hidden states at each layer. We then use this energy score to identify the layer whose hidden state provides the most reliable representation of the input. In detail, the energy score is given by

$$\mathbf{Energy}(h_t^k) = -\operatorname{LogSumExp}[\mathcal{H}(h_t^k)] \tag{F.4}$$

where $\mathcal{H}(h_t^k)$ denote the logits calculated at layer k for predicting token t. The layer $k^* = \arg\min_k \mathbf{Energy}(h_t^k)$ with the lowest score is consequently selected for decoding. An illustration of our method is shown in Figure 2, and Alg. 2 lists the respective Python code.

4 Experiments

The proposed method is demonstrated on three discriminative benchmarks including POPE [11], MME [12] and MMVP [13] and two open-sourced VLMs. We closely follow the protocol conducted in VCD [1] when performing on POPE and MME benchmarks. For MMVP, we follow the evaluation done by [13].

Algorithm 2: Energy-Guided Decoding in Pytorch-like Pseudocode

```
Input  : outputs.hidden_states # List of hidden states from transformer
         model
Output: next_token_scores # Next-token scores derived from the
         minimal-energy hidden state
all_hidden_energy = torch.zeros(1,
 len(outputs.hidden_states)).to("cuda:0") # Initialize tensor to store energy
 values;
for i ← 1 to len(outputs.hidden_states) do
    # Compute logits for current hidden state;
    hidden_logits = self.lm_head(outputs.hidden_states[i])
    # Calculate "negated energy score" over the logits of the last token;
    hidden_energy = LogSumExp(hidden_logits[:, -1, :])
    # Store energy in all_hidden_energy;
    all_hidden_energy[0, i] = hidden_energy
# Find the index of the minimal energy;
max_idx = torch.sort(all_hidden_energy, descending=True).indices[:, 0]
# Compute next-token scores from highest energy state;
next_token_scores = self.lm_head(outputs.hidden_states[max_idx])
```

4.1 Datasets and Evaluation Metrics

POPE Polling-based Object Probing Evaluation (POPE) [11] is a commonly-used benchmark to evaluate the performance of object hallucination [1], [2], [17]. It consists of three datasets including MSCOCO [41], A-OKVQA [20], and GQA [21]. For each dataset, 500 images are sampled with three different sampling strategies including random sampling, popular sampling, and adversarial sampling. Random sampling refers to randomly sample the objects that do not exist in the image. Popular sampling refers to selecting the top half of the most frequent objects in the whole datasets. Adversarial setting is the most difficult configuration, where it first sorts all objects based on their co-occurring frequencies with the ground-truth objects, then selects the top half frequent ones that do not exist in the image. Further, there are 6 questions formulated from each image. In total, there are 27,000 query-answer pairs.

MME The original Multimodal Large Language Model Evaluation (MME) benchmark consists of 10 perception-related tasks and 4 cognition-based tasks.

Datasets	Hallucination-types	# Pairs
POPE-MSCOCO [41]	category	9,000
POPE-AOKVQA [20]	category	9,000
POPE-GQA [21]	category	9,000
MME [12]	category, attribute	240
MMVP [13]	category, attribute, relation	300

Table 2: *Specifications of Hallucination Benchmarks.*

We closely follow [1], [17], [42] to perform hallucination evaluation on the perceptional subtasks. Specifically, the existence and count tasks are employed for object-level hallucination evaluation and the position and color tasks for attribute-level hallucination evaluation. Each image is designed with two questions. The performance is evaluated by the sum of accuracy and accuracy+, where accuracy refers to the proportion of correct answers and accuracy+ refers to the proportion of both questions are answered correctly.

MMVP Multimodal Visual Patterns (MMVP) benchmark [13] consists 150 images with 300 questions. The collected paired images are CLIP-blind meaning that their cosine similarity exceeds 0.95 for CLIP embeddings and less than 0.6 for DINOv2 embeddings. The evaluation of original MMVP benchmark relies on either the GPT-grader or manually comparing the generated responses with the ground truth answers. To provide an accurate evaluation, we select 122 image-questions pairs that share the similar prompt template as POPE [11] and MME [12] meaning the response is either yes or no. Therefore, we can employ the metrics including accuracy and F1 score.

Evaluation Metrics We closely follow the evaluation protocol established by VCD [1] for the POPE benchmark and MME benchmark. Specifically, accuracy and F1 score (i.e., the harmonic mean of precision and recall) are commonly employed to measure the presence of hallucinations. Unlike VCD [1] , which simply shows the values of the yes ratio, we choose to depict the gap between the the predicted and the expected yes ratio, which reflects the degree of bias more directly. To be specific, the yes ratio gap is defined as

$$\Delta_{\text{gap}} = \left| \frac{\text{\# of answers with yes}}{\text{\# of total questions}} - 0.5 \right|, \tag{F.5}$$

where $|\cdot|$ denotes the absolute value and 0.5 represents the expected yes ratio because the dataset is balanced. For the MME benchmark, we follow VCD [1] to report the sum of accuracy(i.e., the number of correct answers over the total questions) and accuracy+ (the number of correctly answering both questions given one image over the total number of images) as the final score. For the MMVP benchmark, we report the same metrics as POPE [11].

4.2 Models and Baselines

VLM backbones Two recent and competitive VLMs including LLaVA-1.5 [4] and InstructBLIP [43] are employed to evaluate the performance of hallucination mitigation. Specifically, InstructBLIP [43] employs the Q-former to extract instruction-aware visual features from the output embeddings of the frozen image encoder. LLaVA-1.5 [4] simply utilizes a Multilayer perceptron (MLP) layer to align the visual feature and text feature. They all employ Vicuna-7B [44] as the language decoder. The template for query VLMs is *Is there a {} in the image?* for all benchmarks as conducted in VCD [1].

Decoding baselines To ensure reproducibility, we use greedy search as the baseline decoding method. We also include two training-free methods designed for mitigating object hallucination, i.e., VCD [1] and HALC [17]. Because of the high computational demand of OPERA [2] its results is included in the supplementary material. For the MME and MMVP benchmarks, we also include regular decoding as an additional baseline. We use their suggested hyperparameters for VCD [1] and HALC [17].

4.3 Experimental Results

POPE results The results in terms of accuracy, F1 score, and yes ratio gap on POPE benchmark with three datasets including MSCOCO [41], A-OKVQA [20], and GQA [21] are presented in Table 3. LLaVA-1.5 [4] and InstructBLIP [43] are employed as the VLM backbones. First, it is worthwhile to note that our method consistently obtains the highest accuracy and the lowest yes ratio gap on two datasets including the A-OKVQA [20] and GQA [21] across three different POPE settings with LLaVa-1.5 as the VLM backbone. Specifically, our method outperforms the baseline method greedy with a large margin up to 10.2% in terms of accuracy and 5.31% in terms of

F1 score. More importantly, when the POPE setting is changed from *random* setting to *adversarial* setting meaning when the task difficulty progressively increases, our method maintains the performance gain in terms of both accuracy and F1 score. Further, the effectiveness of our method in terms of accuracy remains when using a less advancing VLM, i.e., InstructBLIP [43]. Additionally, we further visualize the confidence of saying " yes" and saying " no" before and after using energy-guided decoding in Figure 4. It is evident that greedy decoding (first row) tends to be more confident in saying "yes" than in saying "no". More importantly, the confidence in saying "no" decreases even further as the setting shifts from random to adversarial. In contrast, our method (energy-guided decoding) maintains a similar level of confidence in both saying "yes" and "no" even as the task becomes more difficult, transitioning from a random to an adversarial setting.

MME-subset results Despite MME [12] has a relatively small dataset size, it covers both category level and attribute level data, allowing us to evaluate our method across there levels. We closely follow VCD [1] nd conduct evaluations on the MME subset. The results, including four baselines across two architectures, are presented in Table 4. The reported scores represent the sum of accuracy(i.e., the number of correct answers over the total questions) and accuracy+ (the number of instances in which both questions associated with an image are answered correctly, over the total number of images). Our method demonstrates effectiveness in mitigating hallucinations at both the category and attribute levels. Specifically, it effectively reduces hallucinations related to *Count* at the category level and *Color* at the attribute level. This effectiveness suggests that our method (energy-guided decoding) could address the inherent biases built in language decoder. In contrast, all baseline methods obtain relatively lower *Position* score under two different VLMs, indicating that the VLMs are incapable at reasoning tasks. However, LLaVa-1.5 equipped with energy-guided decoding (our method) achieves a noticeable improvement at the *Postion* score.

MMVP-subset results The original MMVP requires either manually checking the generated response or GPT-grader for evaluation. To facilitate the time for evaluation, we select a subset of image-question pairs that require to answer yes or no for evaluation. The same metrics utilized in POPE [11]

Datasets	*Settings*	**Decoding**	**LLaVA-1.5** Accuracy↑	F1 Score↑	$\Delta_{\text{gap}}\downarrow$	**InstrcutBLIP** Accuracy↑	F1 Score↑	$\Delta_{\text{gap}}\downarrow$
MSCOCO	*Random*	Greedy	**89.37**	**89.33**	**0.37**	**90.17**	**89.86**	3.03
		VCD	84.83	85.30	3.17	84.47	84.38	**0.53**
		HALC	89.30	89.25	0.50	89.73	89.52	2.07
		Energy (Ours)	87.50	86.22	9.30	86.80	85.10	11.40
	Popular	Greedy	86.00	86.41	3.00	83.47	**84.05**	**3.67**
		VCD	81.77	82.81	6.10	77.73	79.12	6.67
		HALC	**86.10**	**86.47**	**2.70**	82.30	83.20	5.37
		Energy (Ours)	85.80	84.63	7.60	**83.70**	82.22	8.31
	Adversarial	Greedy	79.10	80.96	9.77	80.67	**81.82**	**6.33**
		VCD	76.17	78.73	12.03	75.87	77.94	9.40
		HALC	79.27	81.05	9.40	79.47	80.99	8.00
		Energy (Ours)	**82.90**	**82.03**	**4.83**	**82.17**	80.90	6.64
A-OKVQA	*Random*	Greedy	85.70	86.90	9.17	**89.13**	**89.50**	**3.47**
		VCD	80.77	82.85	12.17	83.23	84.23	6.30
		HALC	85.80	86.98	9.07	88.27	88.85	5.20
		Energy (Ours)	**88.60**	**88.30**	**2.6**	89.07	88.44	5.40
	Popular	Greedy	79.90	82.52	14.97	79.57	81.92	13.03
		VCD	76.47	79.83	16.67	76.87	79.73	14.13
		HALC	79.97	82.56	14.9	78.20	81.09	15.27
		Energy (Ours)	**84.67**	**84.87**	**1.33**	**84.03**	**83.97**	**0.37**
	Adversarial	Greedy	69.07	75.41	25.80	71.43	76.42	21.17
		VCD	68.47	74.54	23.87	69.23	74.28	19.63
		HALC	69.23	75.51	25.63	70.33	75.91	23.13
		Energy (Ours)	**77.40**	**79.19**	**8.59**	**76.70**	**78.22**	**6.96**
GQA	*Random*	Greedy	85.77	87.09	10.23	**86.90**	**87.30**	**3.17**
		VCD	81.33	83.34	12.07	80.90	82.07	6.49
		HALC	85.90	87.19	10.10	85.97	86.55	4.37
		Energy (Ours)	**89.37**	**89.19**	**1.63**	86.53	85.54	6.87
	Popular	Greedy	74.73	79.16	11.27	76.37	79.21	13.7
		VCD	71.53	76.82	22.8	73.00	76.32	14.0
		HALC	74.87	79.25	21.13	74.50	77.99	15.83
		Energy (Ours)	**82.53**	**83.40**	**5.2**	**80.27**	**80.15**	**0.60**
	Adversarial	Greedy	69.43	75.85	26.57	71.50	75.96	18.56
		VCD	68.97	75.14	24.83	69.10	73.85	18.16
		HALC	69.53	75.91	26.47	69.70	74.88	20.63
		Energy (Ours)	**79.63**	**81.16**	**8.09**	**76.57**	**77.27**	**3.10**

Table 3: *Results on POPE benchmark* with LLaVa-1.5 [45] and InstrucBLIP [43] as the VLMS backbones. The prompt used for all methods is *"Is there a {} in the image?"*. Higher accuracy and F1 score indicate better performance and fewer hallucinations. Lower yes ratio gap, Δ_{gap} (F.5), implies the model is better calibrated. The best performing method within each setting in **bold**, the 2nd best is underlined.

Models	Decoding	Category-level		Attribute-level		
		Existence↑	*Count*↑	*Position*↑	*Color*↑	Total Scores↑
LLaVA1.5	Regular	180.00	86.67	75.00	135.00	476.67
	Greedy	190.00	110.00	96.67	135.00	531.67
	VCD	170.00	103.33	100.00	130.90	504.23
	HALC	190.00	110.00	96.67	135.00	531.67
	Energy (Ours)	**195.00**	**148.33**	**128.33**	**170.00**	**641.67**
InstructBLIP	Regular	183.33	101.67	**85.00**	88.33	458.33
	Greedy	185.00	93.33	76.67	110.00	465.00
	VCD	173.33	91.67	78.33	88.33	431.66
	HALC	**185.00**	81.67	70.00	110.00	446.67
	Energy (Ours)	180.00	**146.67**	56.67	**140.00**	**523.34**

Table 4: *Results on the subset of MME [12].* Regular decoding denotes direct sampling, whereas Energy refers to sampling from the predictive distribution derived from the hidden states with minimal energy score. The prompt used for all methods is *"Is there a {} in the image?"*. The best performing method within each setting in **bold**.

including accuracy, F1 score, and yes ratio gap are reported. One can see from Table 5 that energy-guided decoding (our method) achieves the best performance in terms of both accuracy and F1 score with the lowest yes ratio gap.

Model	**Decoding**	Accuracy↑	Precision	Recall	F1 Score↑	$\Delta_{\text{gap}}\downarrow$
LLaVA-1.5	Regular	59.02	55.91	85.25	67.53	26.23
	Greedy	57.38	54.46	90.16	67.90	32.79
	VCD	60.66	56.19	96.72	71.08	36.07
	Energy (Ours)	**64.75**	62.50	73.77	**67.67**	**9.02**
InstructBLIP	Regular	55.74	55.07	62.30	58.46	6.55
	Greedy	63.93	61.04	77.09	68.12	13.15
	VCD	52.46	52.05	62.30	56.72	9.84
	Energy (Ours)	**64.75**	63.24	70.49	**66.67**	**5.74**

Table 5: *Results on MMVP dataset.* Higher accuracy and F1 score indicate better performance and fewer hallucinations. Lower yes ratio gap, Δ_{gap} (F.5), implies the model is better calibrated. The best entries within each setting are in **bold**.

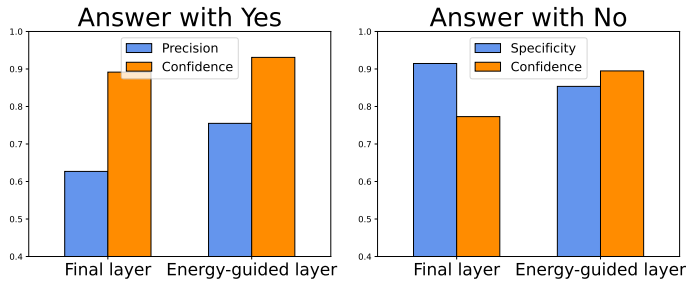

Figure 5: *Accuracy vs. Confidence* for answers with yes (left) and the ones with no (right), using hidden states from the last layer and the ones with mimimum energy for decoding, respectively. The GQA dataset with *adversarial* setting is utilized along with *greedy* decoding. LLaVa-1.5 [4] is utilized the VLM backbone.

4.4 Ablation Studies

Accuracy vs. confidence We visualize the accuracy and confidence for answers with "yes" and "no" in Fig. 5. The accuracy of answers with "yes" and answers with "no" can be calculated as precision and specificity, respectively. The confidence of each answer is measured by the predictive probability of the corresponding token, i.e., the probability distribution after the Softmax layer. Therefore, the confidence shown in Fig. 5 is the average confidence of answers with "yes" and "no", respectively. One can see that the gap between precision and averaged confidence of answers with "yes" is reduced after dynamically selecting the layer based on the corresponding energy score. Similarly, the gap between specificity and averaged confidence of answers with "no" is also reduced after applying energy-guided decoding. It indicates that our method (energy-guided layer) provides a better calibrated answer compared to the final layer. Comparisons for other datasets and models can be founded in supplementary material.

Energy score visualization We empirically observe the hidden states selected by the energy-guided decoding mostly come from the second last layer. Therefore, we visualize the energy score across each layer in Fig 6 with LLaVa-1.5 as the VLM backbone. One can see the energy score calculated from the penultimate layer is generally lower than other layers, indicating that the

corresponding hidden states is more reliable than that from other layers. Visualizations for other datasets and models can be founded in supplementary material.

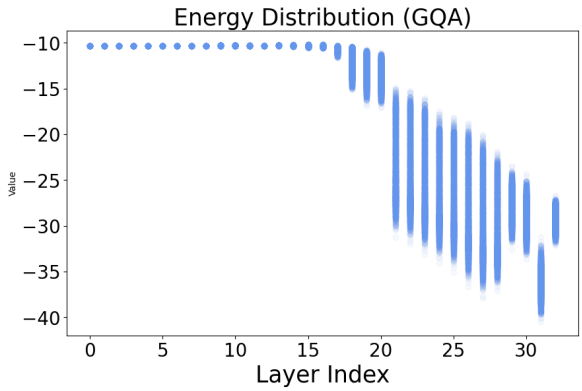

Figure 6: *Energy score distribution* with LLaVa-1.5 [4] as the VLM backbone. The GQA dataset with *adversarial* setting is utilized for evaluation.

5 Conclusion and Discussion

In this work, we empirically observe a notable bias in terms of "*yes*" ratio within VLMs when utilizing the hidden states from the final layer for decoding. Moreover, the"*yes*" ratio biases increase when the tasks are becoming more challenging (e.g., from *random* setting to *adversarial* setting). Inspired by "logit lens" [32], we project the hidden states extracted from each layer to the "unembedding matrix" of the language decoder to obtain multiple logit distributions. Further, we utilize the energy score as a metric to identify the most reliable hidden states for the subsequent decoding procedure. The proposed energy-guided decoding is simple and effective, leading to improved performance in terms of accuracy and F1 score, with a reduced "*yes*" ratio gap. While we primarily focus on the "*yes*" ratio bias in VLMs, which may originate from the language prior [14]. We hypothesize that similar biases and potential issues in LLMs are also expected to transfer to VLMs, as demonstrated by the "yes" ratio bias empirically observed in our study.

6 Supplementary Material

I Experimental results

In this section, we show detailed hyperparameter settings and results, including accuracy, precision, recall, F1 score, Yes ratio and yes ratio gap for three baseline methods including VCD [1] , HALC [17] , and OPERA [2]. All experiments are running on an NVIDIA GeForce RTX 3090 GPU, CUDA 11.4 + PyTorch 2.0.0.

Datasets

We utilize the data list provided by VCD [1] for the POPE [11] benchmark, and can be found here[2]. For the MMVP [13] enchmark, we first correct one error in the dataset, i.e., the answers for image-question pairs 279 and 280 are incorrect. Besides, we select the image-question pairs that requires to answer "yes" or "no" and report the same metrics as for the POPE [11] benchmark. The template for query VLMs is *Is there a* {} *in the image?* for all benchmarks as conducted in VCD [1].

Hyperparameter settings

We conduct the experiments with the hyperparameter setting implemented in HALC [17] to ensure the fair comparison. We mainly focus on the discriminative tasks, *i.e.*, only the first word is taken consideration for the evaluation. Therefore, we set the number of maximum tokens to be 16 to enable faster inference. The temperature is set to 1 for all experiments.

Detailed results on POPE benchmark

The results for hallucination mitigation on the POPE [11] benchmark with LLaVA-1.5 [4] and InstructBLIP [43] as the vision-language model (VLM) backbone are presented in Table 8 and Table 9, respectively. We also include results for OPERA [2] that necessitates the computationally costly beam search decoding. One can see that energy-guided decoding (our method) consistently obtains the best results in terms of accuracy, F1 score, and yes ratio gap across two datasets including A-OKVQA [20] and GQA [21] with three

[2]https://github.com/DAMO-NLP-SG/VCD/tree/master/experiments/data/POPE

Decoding methods	Parameters	Value
VCD [1]	Amplification Factor α	1
	Adaptive Plausibility Threshold β	0.1
	Noise Step	500
HALC [17]	Contrast weight α	0.05
	JSD Candidate number k	6
	Number of Sampled FOVs n	4
	Exponential Growth factor λ	0.6
	Adaptive Plausibility Threshold β	0.1
OPERA [2]	Self-attention Weights Scale Factor θ	50
	Attending Retrospection Threshold	15
	Beam Size	3
	Penalty weights	1

Table 6: *Hyperparameter settings* for the baseline methods.

different configurations including *random*, *popular*, and *adversarial* when utilizing LlaVA-1.5 [4] as the VLM backbone. Specifically, the average accuracy improvement is 4.37% and the average yes ratio gap reduction is 8.11% compared to vanilla greedy decoding. OPERA [2], as one of the competitive baseline method, is inferior to our method on GQA dataset across three settings in terms of accuracy, F1 score, and yes ratio gap. Particularly, our method outperforms OPERA [2] with a margin 1.37% and 3.97% in terms of accuracy and yes ratio gap, respectively. The results with InstructBLIP [43] as the VLM backbone follow a similar, but less pronounced pattern.

Detailed results on the MMVP benchmark

The results of hallucination mitigation on the subset of the MMVP [13] benchmark with LLaVA-1.5 [4] and InstructBLIP [43] as the vision-language model (VLM) backbone are presented in Table 7. We also include the results in terms of yes ratio. One can see that, energy-guided decoding (our method) consistently obtains the best accuracy and yes ratio gap across two architectures. Specifically, the average yes ratio gap is reduced by a margin of 15.59% compared to the greedy decoding.

Model	**Decoding**	Accuracy↑	Precision	Recall	F1 Score↑	Yes ratio	$\Delta_{\text{gap}}\downarrow$
LLaVA-1.5	Regular	59.02	55.91	85.25	67.53	76.23	<u>26.23</u>
	Greedy	57.38	54.46	90.16	67.90	82.79	32.79
	VCD	<u>60.66</u>	56.19	96.72	**71.08**	86.07	36.07
	Energy (Ours)	**64.75**	62.50	73.77	<u>67.67</u>	59.02	**9.02**
InstructBLIP	Regular	55.74	55.07	62.30	58.46	56.55	<u>6.55</u>
	Greedy	<u>63.93</u>	61.04	77.09	**68.12**	63.15	13.15
	VCD	52.46	52.05	62.30	56.72	59.84	9.84
	Energy (Ours)	**64.75**	63.24	70.49	<u>66.67</u>	55.74	**5.74**

Table 7: *Results on MMVP dataset.* Higher accuracy and F1 score indicate better performance and fewer hallucinations. Lower yes ratio gap (Δ_{gap}) implies the model is better calibrated. The best entries within each setting are in **bold**, the 2nd is <u>underlined</u>.

II Accuracy vs. confidence

In this section,we visualize the accuracy and confidence for answers with " *yes* " and " *no*" in Fig. 7 for three datasets including MSCOCO [41], A-OKVQA [20], and GQA [21] with the *adversarial* setting. The accuracy of answers with "*yes*" and answers with "*no*" can be calculated as precision and specificity, respectively. The confidence of each answer is measured by the predictive probability of the corresponding token. Additionally, the confidence shown in Fig. 7 is the average confidence of answers with " *yes* " and " *no*", respectively. One can see from Fig. 7 that energy-guided decoding generally narrows the gap between accuracy and confidence, i.e., the gap between precision and confidence for answer with "*yes*" and the gap between specificity and confidence for answer with "*no*". That is to say, energy-guided decoding provides better calibrated answers.

III Energy score distribution

We visualize the energy distribution at every layer in Fig. 8 for three datasets including MSCOCO [41], A-OKVQA [20], and GQA [21] with *adversarial* setting. Each dataset consists of 3000 pairs of image-questions. It can be seen that the energy score induced by the penultimate layer is generally the lowest, and that this layer is predominantly utilized for the subsequent decoding process.

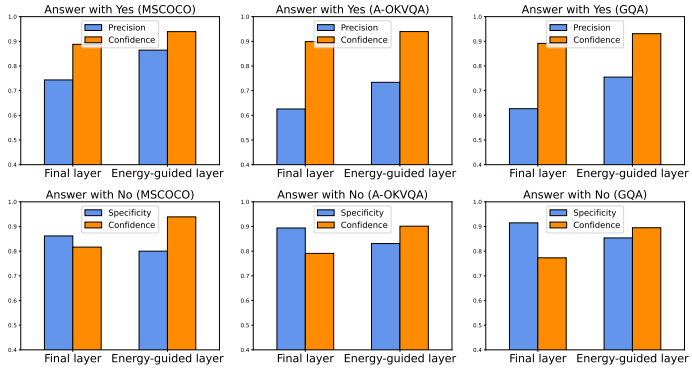

Figure 7: *Accuracy vs. Confidence* for answers with "*yes*" (top row) and the ones with "*no*" (bottom row), using hidden states from the last layer and the ones with minimum energy for decoding, respectively. Three datasets including MSCOCO, A-OKVQA, and GQA with *adversarial* setting are utilized along with *greedy* decoding. LLaVA-1.5 [4] is utilized as the VLM backbone.

IV Yes ratio transfer under regular sampling

We study the yes ratio transfer when using regular sampling. The experimental setting is similar to the ones using greedy sampling. Specifically, the evaluation covering two scenarios—one that includes visual input and one without visual input and the results are shown in Fig. 9. Each number in the "confusion matrix" represents the number of samples (image-question pairs) that overlap between cases with and without visual inputs. For instance, 1207 in the left plot represents the number of image-question pairs that consistently generate the answer "*yes*" regardless of whether visual inputs are provided. One can see from Fig. 9 that the VLM exhibits a similar pattern as using greedy decoding, i.e., the model tends to answer "yes" when the VQA tasks are becoming more difficult (from *random* to *adversarial*) and the "yes" ratio is initially high without visual input, suggesting models are biased toward "yes" due to language priors in this dataset.

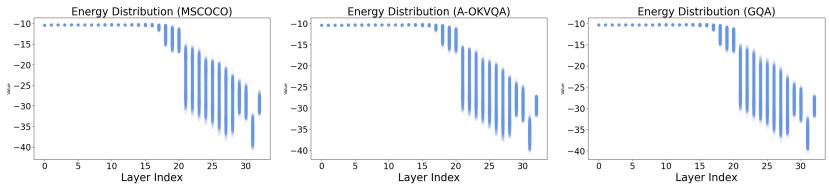

Figure 8: *Energy distribution* at each layer with LLaVa-1.5 [4] as the VLM backbone. Three datasets including MSCOCO [41], A-OKVQA [20], and GQA [21] with *adversarial* setting are utilized along with *greedy* decoding. LLaVA-1.5 [4] is utilized as the VLM backbone.

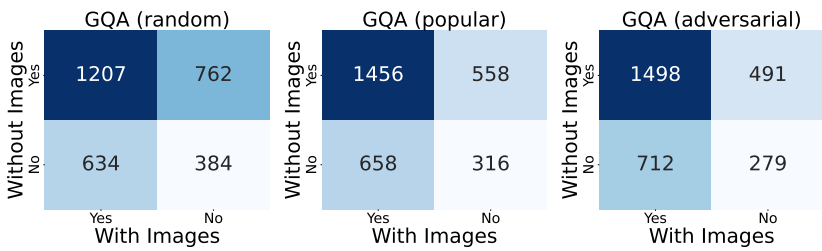

Figure 9: *Transfer of "yes" ratio* from non-visual input to visual inputs using regular decoding. Three settings of POPE-GQA [11] are utilized including random (left column), popular (middle column), and adversarial (right column). LLaVA-1.5 [4] is employed as the VLM backbone.

Dataset	Setting	Decoding	Accuracy↑	Precision	Recall	F1 Score↑	Yes ratio	$\Delta_{\text{gap}}\downarrow$
MSCOCO	*Random*	Greedy	**89.37**	89.66	89.00	**89.33**	49.63	**0.37**
		HALC	89.30	89.70	88.80	89.25	49.5	0.5
		VCD	84.83	82.76	88.00	85.30	53.17	3.17
		OPERA	89.17	92.48	85.27	88.73	46.10	3.90
		Energy (Ours)	87.50	96.07	78.20	86.22	40.70	9.30
	Popular	Greedy	86.00	83.96	89.00	86.41	53.00	3.00
		HALC	86.10	84.25	88.80	86.47	52.70	2.7
		VCD	81.77	78.31	87.87	82.81	56.10	6.10
		OPERA	**86.80**	87.96	85.27	**86.59**	48.47	**1.53**
		Energy (Ours)	85.80	92.22	78.20	84.63	42.4	7.6
	Adversarial	Greedy	79.10	74.34	88.87	80.96	59.77	9.77
		HALC	79.27	74.64	88.67	81.05	59.40	9.40
		VCD	76.17	71.09	88.20	78.73	62.03	12.03
		OPERA	81.20	78.89	85.20	81.92	54.00	6.00
		Energy (Ours)	**82.90**	86.42	78.07	**82.03**	45.17	**4.83**
A-OKVQA	*Random*	Greedy	85.70	80.17	94.87	86.90	59.17	9.17
		HALC	85.80	80.30	94.87	86.98	59.07	9.07
		VCD	80.77	74.75	92.93	82.85	62.17	12.17
		OPERA	88.23	86.09	91.20	**88.57**	52.97	2.97
		Energy (Ours)	**88.60**	90.72	86.00	88.30	47.4	**2.6**
	Popular	Greedy	79.90	73.01	94.87	82.52	64.97	14.97
		HALC	79.97	73.09	94.87	82.56	64.9	14.90
		VCD	76.47	69.85	93.13	79.83	66.67	16.67
		OPERA	83.37	78.85	91.20	84.57	57.83	7.83
		Energy (Ours)	**84.67**	83.77	86.00	**84.87**	51.33	**1.33**
	Adversarial	Greedy	69.07	62.58	94.87	75.41	75.80	25.8
		HALC	69.23	62.71	94.87	75.51	75.63	25.63
		VCD	68.47	62.50	92.33	74.54	73.87	23.87
		OPERA	73.90	67.76	91.20	77.75	67.30	17.30
		Energy (Ours)	**77.40**	73.38	86.00	**79.19**	58.59	**8.59**
GQA	*Random*	Greedy	85.77	79.69	96.00	87.09	60.23	10.23
		HALC	85.90	79.87	96.00	87.19	60.10	10.10
		VCD	81.33	75.24	93.40	83.34	62.07	12.07
		OPERA	88.57	85.47	92.93	89.05	54.37	4.37
		Energy (Ours)	**89.37**	90.70	87.73	**89.19**	48.37	**1.63**
	Popular	Greedy	74.73	67.35	96.00	79.16	71.27	21.27
		HALC	74.87	67.48	96.00	79.25	71.13	21.13
		VCD	71.53	64.79	94.33	76.82	72.8	22.80
		OPERA	79.83	73.64	92.93	82.17	63.10	23.10
		Energy (Ours)	**82.53**	79.47	87.73	**83.40**	55.20	**5.20**
	Adversarial	Greedy	69.43	62.69	96.00	75.85	76.57	26.57
		HALC	69.53	62.77	96.00	75.91	76.47	26.47
		VCD	68.97	62.67	93.80	75.14	74.83	24.83
		OPERA	75.00	68.40	92.93	78.80	67.93	17.93
		Energy (Ours)	**79.63**	75.50	87.73	**81.16**	58.09	**8.09**

Table 8: *Results on POPE benchmark with LLaVA-1.5 [4] as the model.* Higher accuracy and F1 score indicate better performance and fewer hallucinations. Lower yes ratio gap, (Δ_{gap}) implies the model is better calibrated.The best performing method within each setting in **bold**.

Dataset	Setting	Decoding	Accuracy↑	Precision	Recall	F1 Score↑	Yes ratio	$\Delta_{\text{gap}}\downarrow$
MSCOCO	*Random*	Greedy	**90.17**	92.76	87.13	**89.86**	46.97	3.03
		VCD	84.47	84.84	83.93	84.38	49.47	**0.03**
		HALC	89.73	91.45	87.67	89.52	47.93	2.07
		OPERA	89.83	93.71	85.40	89.36	45.57	4.43
		Energy (Ours)	86.80	97.67	75.40	85.10	38.60	11.40
	Popular	Greedy	83.47	81.18	87.13	84.05	53.67	3.67
		VCD	77.73	74.47	84.40	79.12	56.67	6.67
		HALC	82.30	79.17	87.67	83.20	55.37	5.37
		OPERA	**84.67**	84.17	85.40	**84.78**	50.73	**0.73**
		Energy (Ours)	83.70	90.41	75.40	82.22	41.70	8.30
	Adversarial	Greedy	80.67	77.22	87.00	81.82	56.33	6.33
		VCD	75.87	71.77	85.27	77.94	59.40	9.40
		HALC	79.47	75.40	87.47	80.99	58.00	8.00
		OPERA	81.43	79.20	85.27	**82.12**	53.83	**3.83**
		Energy (Ours)	**82.17**	87.09	75.53	80.90	43.37	6.63
A-OKVQA	*Random*	Greedy	89.13	86.60	92.60	**89.50**	53.47	3.47
		VCD	83.23	79.51	89.53	84.23	56.30	6.30
		HALC	88.27	84.66	93.47	88.85	55.20	5.20
		OPERA	**89.57**	88.97	90.33	89.65	50.77	**0.77**
		Energy (Ours)	89.07	93.80	83.67	88.44	44.60	5.40
	Popular	Greedy	79.57	73.45	92.60	81.92	63.03	13.03
		VCD	76.87	70.95	91.00	79.73	64.13	14.13
		HALC	78.20	71.60	93.47	81.09	65.27	15.27
		OPERA	82.67	78.32	90.33	83.90	57.67	7.67
		Energy (Ours)	**84.03**	84.28	83.67	**83.97**	49.63	**0.37**
	Adversarial	Greedy	71.43	65.06	92.60	76.42	71.17	21.17
		VCD	69.23	63.81	88.87	74.28	69.63	19.63
		HALC	70.33	63.90	93.47	75.91	73.13	23.13
		OPERA	74.13	68.23	90.33	77.74	66.20	16.20
		Energy (Ours)	**76.70**	73.43	83.67	**78.22**	56.97	**6.97**
GQA	*Random*	Greedy	86.90	84.70	90.07	87.30	53.17	3.17
		VCD	80.90	77.35	87.40	82.07	56.50	6.50
		HALC	85.97	83.08	90.33	86.55	54.37	4.37
		OPERA	**87.33**	87.23	87.47	**87.35**	50.13	**0.13**
		Energy (Ours)	86.53	92.35	79.67	85.54	43.13	6.87
	Popular	Greedy	76.37	70.70	90.07	79.21	63.70	13.70
		VCD	73.00	67.97	87.00	76.32	64.00	14.00
		HALC	74.50	68.61	90.33	77.99	65.83	15.83
		OPERA	79.77	75.79	87.47	**81.21**	57.70	7.70
		Energy (Ours)	**80.27**	80.63	79.67	80.15	49.40	**0.60**
	Adversarial	Greedy	71.50	65.68	90.07	75.96	68.57	18.57
		VCD	69.10	64.01	87.27	73.85	68.17	18.17
		HALC	69.70	63.95	90.33	74.88	70.63	20.63
		OPERA	74.00	68.91	87.47	77.09	63.47	13.47
		Energy (Ours)	**76.57**	75.02	79.67	**77.27**	53.10	**3.10**

Table 9: *Results on POPE benchmark with InstructBLIP [43].* Higher accuracy and F1 score indicate better performance and fewer hallucinations. Lower yes ratio gap (Δ_{gap}) implies the model is better calibrated. The best performing method within each setting in **bold**, the 2nd is underlined.

References

[1] S. Leng, H. Zhang, G. Chen, *et al.*, "Mitigating object hallucinations in large vision-language models through visual contrastive decoding," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[2] Q. Huang, X. Dong, P. zhang, *et al.*, "Opera: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation," *arXiv:2311.17911*, 2023.

[3] A. Deng, Z. Chen, and B. Hooi, "Seeing is believing: Mitigating hallucination in large vision-language models via clip-guided decoding," *arXiv:2402.15300*, 2024.

[4] H. Liu, C. Li, Y. Li, and Y. J. Lee, "Improved baselines with visual instruction tuning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 26 296–26 306.

[5] OpenAI, *ChatGPT*, `https://openai.com/blog/chatgpt/`, 2023.

[6] A. Awadalla, I. Gao, J. Gardner, *et al.*, "Openflamingo: An open-source framework for training large autoregressive vision-language models," *arXiv:2308.01390*, 2023.

[7] D. Driess, F. Xia, M. S. M. Sajjadi, *et al.*, "Palm-e: An embodied multimodal language model," *arXiv:2303.03378*, 2023.

[8] R. Zhang, J. Han, C. Liu, *et al.*, "Llama-adapter: Efficient fine-tuning of language models with zero-init attention," *arXiv:2303.16199*, 2023.

[9] L. Weng, *Extrinsic hallucinations in llms.* `https://lilianweng.github.io/posts/2024-07-07-`*hallucination*, 2020.

[10] B. et al., "Hallucination of multimodal large language models: A survey," *arXiv:2404.18930*, 2024.

[11] Y. Li, Y. Du, K. Zhou, J. Wang, W. X. Zhao, and J.-R. Wen, "Evaluating object hallucination in large vision-language models," in *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[12] C. Fu, P. Chen, Y. Shen, *et al.*, "Mme: A comprehensive evaluation benchmark for multimodal large language models," *arXiv:2306.13394*, 2023.

[13] S. Tong, Z. Liu, Y. Zhai, Y. Ma, Y. LeCun, and S. Xie, "Eyes wide shut? exploring the visual shortcomings of multimodal llms," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[14] F. Liu, K. Lin, L. Li, J. Wang, Y. Yacoob, and L. Wang, "Aligning large multi-modal model with robust instruction tuning," *arXiv:2306.14565*, 2023.

[15] Y.-S. Chuang, Y. Xie, H. Luo, Y. Kim, J. Glass, and P. He, "Dola: Decoding by contrasting layers improves factuality in large language models," *International Conference on Learning Representations (ICLR)*, 2024.

[16] X. Wang, J. Pan, L. Ding, and C. Biemann, "Mitigating hallucinations in large vision-language models with instruction contrastive decoding," *arXiv:2403.18715*, 2024.

[17] Z. Chen, Z. Zhao, H. Luo, H. Yao, B. Li, and J. Zhou, "Halc: Object hallucination reduction via adaptive focal-contrast decoding," *International Conference on Machine Learning (ICML)*, 2024.

[18] A. Rohrbach, L. Hendricks, K. Burns, T. Darrell, and K. Saenko, "Object hallucination in image captioning," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[19] Y. Zhou, C. Cui, J. Yoon, *et al.*, "Analyzing and mitigating object hallucination in large vision-language models," in *International Conference on Learning Representations (ICLR)*, 2024.

[20] D. Schwenk, A. Khandelwal, C. Clark, K. Marino, and R. Mottaghi, "A-okvqa: A benchmark for visual question answering using world knowledge," *arXiv:2206.01718*, 2022.

[21] D. A. Hudson and C. D. Manning, "Gqa: A new dataset for real-world visual reasoning and compositional question answering," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR*, 2019, pp. 6700–6709.

[22] X. L. Li, A. Holtzman, D. Fried, *et al.*, "Contrastive decoding: Open-ended text generation as optimization," in *the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2023.

[23] Y.-F. Zhang, W. Yu, Q. Wen, *et al.*, "Debiasing large visual language models," *arXiv:2403.05262*, 2024.

[24] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," *arXiv:1904.09751*, 2020.

[25] G. Kobayashi, T. Kuribayashi, S. Yokoi, and K. Inui, "Attention is not only a weight: Analyzing transformers with vector norms," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, 2020.

[26] A. Conmy, A. N. Mavor-Parker, A. Lynch, S. Heimersheim, and A. Garriga-Alonso, "Towards automated circuit discovery for mechanistic interpretability," in *Advances in Neural Information Processing Systems*, 2023.

[27] H. Chefer, S. Gur, and L. Wolf, "Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 397–406.

[28] K. Meng, D. Bau, A. Andonian, and Y. Belinkov, "Locating and editing factual associations in GPT," *Advances in Neural Information Processing Systems*, 2022.

[29] M. Geva, R. Schuster, J. Berant, and O. Levy, "Transformer feed-forward layers are key-value memories," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

[30] M. Geva, A. Caciularu, K. Wang, and Y. Goldberg, "Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space," in *Conference on Empirical Methods in Natural Language Processing*, 2022.

[31] G. Dar, M. Geva, A. Gupta, and J. Berant, "Analyzing transformers in embedding space," in *Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023.

[32] nostalgebraist, "Interpreting gpt: The logit lens.," 2024.

[33] D. Halawi, J.-S. Denain, and J. Steinhardt, "Overthinking the truth: Understanding how language models process false demonstrations," *arXiv:2307.09476*, 2023.

[34] N. Belrose, Z. Furman, L. Smith, *et al.*, "Eliciting latent predictions from transformers with the tuned lens," *arXiv:2303.08112*, 2023.

[35] T. Bolukbasi, K.-W. Chang, J. Zou, V. Saligrama, and A. Kalai, "Man is to computer programmer as woman is to homemaker? debiasing word embeddings," in *Advances in Neural Information Processing Systems*, 2016.

[36] S. Ravfogel, M. Twiton, Y. Goldberg, and R. Cotterell, "Linear adversarial concept erasure," *arXiv:2201.12091*, 2024.

[37] Y. Ouali, A. Bulat, B. Martinez, and G. Tzimiropoulos, "Clip-dpo: Vision-language models as a source of preference for fixing hallucinations in lvlms," *arXiv:2408.10433*, 2024.

[38] W. Liu, X. Wang, J. Owens, and Y. Li, "Energy-based out-of-distribution detection," in *Advances in Neural Information Processing Systems*, 2020.

[39] W. Grathwohl, K.-C. Wang, J.-H. Jacobsen, D. Duvenaud, M. Norouzi, and K. Swersky, "Your classifier is secretly an energy based model and you should treat it like one," in *International Conference on Learning Representations (ICLR)*, 2020.

[40] J. Burapacheep and Y. Li, "Your classifier can be secretly a likelihood-based ood detector," *arXiv:2408.04851*, 2024.

[41] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, "Microsoft coco: Common objects in context," *arXiv:1405.0312*, 2015.

[42] S. Yin, C. Fu, S. Zhao, *et al.*, "Woodpecker: Hallucination correction for multimodal large language models," *arXiv:2310.16045*, 2023.

[43] W. Dai, J. Li, D. Li, *et al.*, "InstructBLIP: Towards general-purpose vision-language models with instruction tuning," in *Advances in Neural Information Processing Systems*, 2023.

[44] W.-L. Chiang, Z. Li, Z. Lin, *et al.*, *Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality*, `https://lmsys.org/blog/2023-03-30-vicuna/`, 2023.

[45] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *arXiv:2304.08485*, 2023.