



Future-Oriented Navigation: Dynamic Obstacle Avoidance with One-Shot Energy-Based Multimodal Motion Prediction

Downloaded from: <https://research.chalmers.se>, 2026-02-27 11:10 UTC

Citation for the original published paper (version of record):

Zhang, Z., Hess, G., Hu, J. et al (2025). Future-Oriented Navigation: Dynamic Obstacle Avoidance with One-Shot Energy-Based Multimodal Motion Prediction. *IEEE Robotics and Automation Letters*, 10(8): 8043-8050.
<http://dx.doi.org/10.1109/LRA.2025.3575969>

N.B. When citing this work, cite the original published paper.

© 2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, or reuse of any copyrighted component of this work in other works.

Future-Oriented Navigation: Dynamic Obstacle Avoidance with One-Shot Energy-Based Multimodal Motion Prediction

Ze Zhang, Georg Hess, Junjie Hu,
Emmanuel Dean, Lennart Svensson, and Knut Åkesson,

Abstract—This paper proposes an integrated approach for the safe and efficient control of mobile robots in dynamic and uncertain environments. The approach consists of two key steps: one-shot multimodal motion prediction to anticipate motions of dynamic obstacles and model predictive control to incorporate these predictions into the motion planning process. Motion prediction is driven by an energy-based neural network that generates high-resolution, multi-step predictions in a single operation. The prediction outcomes are further utilized to create geometric shapes formulated as mathematical constraints. Instead of treating each dynamic obstacle individually, predicted obstacles are grouped by proximity in an unsupervised way to improve performance and efficiency. The overall collision-free navigation is handled by model predictive control with a specific design for proactive dynamic obstacle avoidance. The proposed approach allows mobile robots to navigate effectively in dynamic environments. Its performance is assessed across various scenarios that represent typical warehouse settings. The results demonstrate that the proposed approach outperforms other existing dynamic obstacle avoidance methods.

Index Terms—Human-aware motion planning, collision avoidance, deep learning methods.

I. INTRODUCTION

FACTORY, warehouse, and laboratory layouts are subject to changes, and the presence of humans and manually driven vehicles further complicates motion planning for Autonomous Mobile Robots (AMRs). To ensure collision avoidance, AMRs must anticipate interactions with static and dynamic objects, though the unpredictability of humans poses significant challenges. A prevalent strategy is to halt the robot when obstacles are detected [1], risking downtime if obstructions persist or deadlocks occur. This study addresses collision-free navigation for AMRs in industrial settings but can be generalized to other settings with similar traffic dynamics.

Dynamic Obstacle Avoidance (DOA) has been a key research area since the advent of AMRs, with an initial emphasis on safety. This often leads to overly cautious behavior, exemplified by the Freezing Robot Problem (FRP) [2], where the planner deems all paths and stops the robot. Advancements

Manuscript received September 30, 2024; revised April 27, 2025 and February 17, 2025; Accepted: May 21, 2025.

This paper was recommended for publication by Gentiane Venture upon evaluation of the Associate Editor and Reviewers' comments.

This work is supported by the AIHURO project (Vinnova 2022-03012). The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council (grant agreement no. 2022-06725).

All authors are with Chalmers University of Technology, 41296 Gothenburg, Sweden ({zhze, georghe, junjiej, deane, lennart.svensson, knut}@chalmers.se).

Digital Object Identifier (DOI): see top of this page.

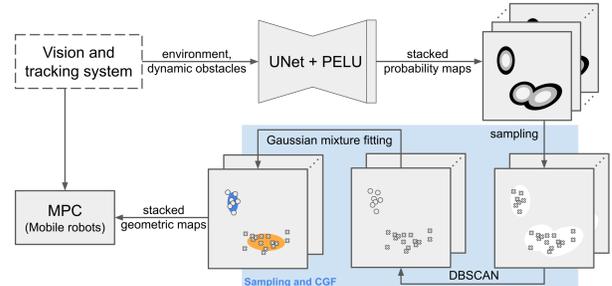


Fig. 1. Pipeline of the proposed obstacle avoidance approach, assuming environmental information is provided by a bird's-eye-view vision system.

in perception and computation technologies enable robots to rapidly and accurately gather extensive environmental data and coordinate actions, improving the ability to implement effective decision-making strategies. Recent research has improved motion planning efficiency and addressed FRP by incorporating human movements and intentions [3], [4], especially in social robot navigation [5]. In traditional model-based approaches, the Constant Velocity Model (CVM) and velocity-based strategies are widely used for predicting human motion in navigation frameworks [3], [6], [7]. Additionally, predefined kinematics-based [8] and interaction-based [9] models offer more detailed insights for motion planners. Recently, learning-based models have become the primary motion predictor in most prescient navigation frameworks due to their high flexibility and ability to interpret complex and uncertain interactions and environments [4], [9]–[11].

This work presents a navigation framework for multiple AMRs in dynamic environments, tailored for industrial transport tasks, incorporating multimodal motion prediction, prediction grouping, static obstacle avoidance, proactive DOA, and fleet collision avoidance. The approach integrates a data-driven motion predictor with a predictive trajectory planner. Specifically, an energy-based model predicts possible future motions of dynamic obstacles as probability maps, which are used by distributed Model Predictive Control (MPC) for safe navigation. Multiple tests are designed to evaluate the method's performance in a comparative study with state-of-the-art approaches. The main contributions are threefold:

- Introducing a fast and stable energy-based learning approach for multimodal motion prediction enabling improved integration with downstream planning and control.
- Improving MPC-based motion planner for effective, safe, and efficient fleet control in dynamic environments.
- Evaluating the framework under various conditions and scenarios, with and without cooperative pedestrians.

II. RELATED WORK

Motion prediction of dynamic obstacles is crucial for the downstream motion planning of AMRs. Early CVM-based methods rely solely on velocity information and have been proven effective [12], but struggle with human behavior's uncertainty. Other model-based approaches, such as social force [13], reciprocal velocity obstacles [14], and other kinematic models [8], also fail to capture the multimodality required to address this uncertainty. Advancements in deep learning have enabled the effective capture of environmental contexts and agent interactions while supporting uncertainty and multimodality. Techniques such as social pooling [15], [16] model pedestrian interactions, Convolutional Neural Networks (CNNs) enhance context perception [17]–[19], attention mechanisms [20] provide a comprehensive and global understanding, and inverse reinforcement learning [21] encodes environmental information as rewards.

Reinforcement learning has been applied to AMR motion planning and obstacle avoidance [3], [22], providing flexibility but facing instability and safety challenges. Field-based methods [7] have been developed to generate smooth motion fields around obstacles. However, these methods fail to address the inherent uncertainty in the environment. Industrial transport robots require stability, flexibility, interpretability, and predictive capabilities. As an optimization-based method, MPC is well-suited for DOA due to its constraint handling and predictive receding horizon. Combining MPC with deep learning for motion prediction improves accuracy and effectiveness in dynamic environments [9]–[11], [23].

This work follows a similar modular structure as in [11] but with two key differences, including a faster and more stable neural network for multimodal motion prediction and an extended MPC controller to achieve fleet collision avoidance [24]. A novel Energy-Based Learning (EBL) strategy is proposed to apply the new neural network architecture.

III. PROBLEM FORMULATION

This work addresses two coherent challenges: AMR controller design for DOA and motion prediction of dynamic obstacles, focusing on 2D spaces and discrete-time domains.

Collision-free Motion Planning: Let the discrete motion model for the i -th AMR is $\mathbf{s}_{k+1}^{(i)} = f(\mathbf{s}_k^{(i)}, \mathbf{u}_k^{(i)})$ in a fleet, where $\mathbf{s}_k^{(i)}$ and $\mathbf{u}_k^{(i)}$ represent its state and action at time step k . The aim is to plan its trajectory along a given reference path while avoiding collisions with obstacles or other robots and adhering to its physical limitations. For simplicity, the identification index (i) is omitted when discussing a single robot for all variables. Static obstacles are assumed to be polygonal. Their occupied area, $\mathcal{O} = \cup_j \mathcal{O}^{(j)}$, is the union of individuals, where each set is a closed intersection of half-spaces. Dynamic obstacles are modeled as ellipses and the corresponding area \mathcal{D}_k may vary over time. Similarly, $\mathcal{D}_k = \cup_j \mathcal{D}_k^{(j)}$. At any time, a robot at position \mathbf{p}_k must be outside any obstacles: $\forall k, \mathbf{p}_k \notin \mathcal{O} \cup \mathcal{D}_k$. Obstacles are inflated by the size of the robot, and robots are regarded as points [11]. **Motion Prediction:** To consider potential collisions with dynamic obstacles, AMRs should have access to the estimated

future positions of obstacles. Let the predictive horizon be N , AMRs require information on other dynamic objects from the current instant to future N steps. For dynamic obstacle i , the task is to predict its future states $\{\mathbf{z}_{t_k}^{(i)}\}_{t_k=k}^{k+N}$. Since future motion can be multimodal, multiple hypotheses of each future position should be available. For a maximal number M of modes at each time step for each dynamic obstacle, the desired prediction is notated as $\{\{\mathbf{z}_{t_k}^{(i)}\}_{m=1}^M\}_{t_k=k}^{k+N}$.

IV. FUTURE-ORIENTED MODEL PREDICTIVE CONTROL

This section outlines the formulation of the MPC problem incorporating DOA with consideration of multiple futures.

A. Dynamic Obstacle Avoidance with Multiple Futures

A dynamic obstacle with M possible futures can be viewed as M potential obstacles, each with a different probability of occurrence. Assuming accurate predictions, a sufficient condition for an AMR to avoid the obstacle within the horizon is that it avoids collisions with all potential obstacles at the corresponding time steps, as in Eq. (1), where ${}^m\hat{\mathcal{D}}_{t_k}$ is the predicted occupied area of obstacle m at t_k . A positive integer set from a to b is denoted as $\mathbb{N}_{[a,b]}$.

$$\forall t_k \in \mathbb{N}_{[k,k+N]}, m \in \mathbb{N}_{[1,M]}, \mathbf{p}_{t_k} \notin {}^m\hat{\mathcal{D}}_{t_k}. \quad (1)$$

This condition can be overly restrictive, especially with large M , as potential obstacles may occupy the entire drivable area, rendering the MPC problem infeasible. Retaining only predicted futures with high probabilities provides limited mitigation of this issue. Thus, two solutions are introduced in this work to address this.

Firstly, rather than enforcing predictive DOA as a hard constraint, a flexible alternative is to add it into the objective as a soft constraint, as shown in Eq. (2) for $t_k \in \mathbb{N}_{[k,k+N]}$. Here, function $\iota_D(\cdot)$ [11] evaluates the distance between the robot and the potential obstacle, β_{t_k} is the weight corresponding to time, and α_m is mode-dependent.

$$J_{\mathcal{D}}(t_k) = \beta_{t_k} \cdot \left[\sum_{m=1}^M \alpha_m \cdot \iota_D(\mathbf{p}_{t_k}, {}^m\mathcal{D}_{t_k}) \right] \quad (2)$$

The mode weight α_m can be determined by the probability of each mode's occurrence, reflecting the intrinsic uncertainty. The time weight β_{t_k} can be chosen such that a larger t_k corresponds to a smaller weight. The second idea is to group predictions at each step based on their proximity [23]. This strategy considers occupied areas collectively rather than focusing on individuals, which not only reduces the computation burden for MPC but can also alleviate the FRP.

B. Model Predictive Control Formation

The objective of MPC is composed of three terms, including the reference deviation term J_R , soft DOA term $J_{\mathcal{D}}$, and fleet-collision-avoidance term [24]. For a robot at time k ,

$$J_R(k) = \|\mathbf{s}_k - \tilde{\mathbf{s}}_k\|_{\mathcal{Q}_s}^2 + \|\mathbf{u}_k - \tilde{\mathbf{u}}_k\|_{\mathcal{Q}_u}^2 + \|\mathbf{u}_k - \mathbf{u}_{k-1}\|_{\mathcal{Q}_a}^2, \quad (3)$$

$$J_{\mathcal{D}}(k) = \sum_{j=1}^{n_d} \|\iota_D(\mathbf{p}_k | \mathbf{q}_k^{(j)}, \boldsymbol{\sigma}_k^{(j)})\|_{\mathcal{Q}_D}^2, \quad (4)$$

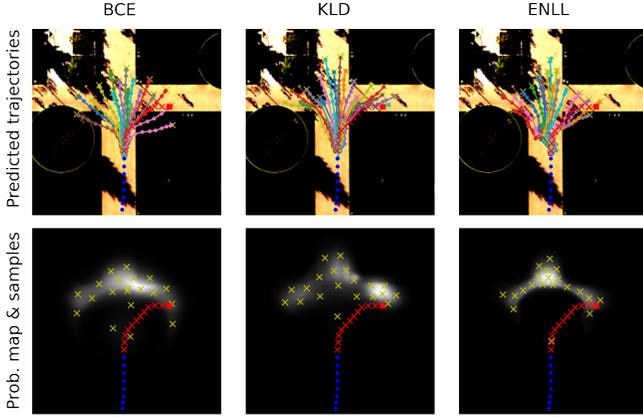


Fig. 2. Training results on the Stanford Drone Dataset [25] with different learning strategies based on the Y-Net [18] architecture. Blue dots indicate the historical trajectory of the target while red crosses are the ground-truth future trajectory. The probability maps display the predicted probabilities for the last future step. Yellow crosses are samples from the maps.

where \tilde{s}_k is the reference state, \tilde{u}_k is the reference action, n_d is the number of dynamic obstacles, $\mathbf{q}_k^{(j)}$ is the position of obstacle j , $\sigma_k^{j'} = \sigma_k^j + r_{\text{extra}}$ is the obstacle axes based on the original one σ_k^j padded by a safe margin r_{extra} , and all \mathbf{Q} variables are penalty parameters. For the fleet collision avoidance between the ego robot i and another robot j ,

$$J_F(j) = \max \left[0, \mathbf{Q}_f \cdot \left(d_{\text{fleet}} - \|\mathbf{p}_k^{(i)} - \mathbf{p}_k^{(j)}\| \right)^2 \right], \quad (5)$$

where d_{fleet} is the safe distance between robots. The overall MPC problem for mobile robot i is formulated as (omit i if there is no ambiguity)

$$\min_{\mathbf{u}_{k:k+N-1}} \sum_{t_k=k}^{k+N-1} \left[J_R(t_k) + J_O(t_k) + J_D(t_k) + \sum_{j=1, j \neq i}^{n_r} J_F(j) \right], \quad (6)$$

$$\text{s.t. } \mathbf{s}_{t_k+1} = f(\mathbf{s}_{t_k}, \mathbf{u}_{t_k}), \forall t_k \in \mathbb{N}_{[k, k+N-1]}, \quad (7)$$

$$\mathbf{u}_{t_k} \in [\mathbf{u}_{\min}, \mathbf{u}_{\max}], \forall t_k \in \mathbb{N}_{[k, k+N-1]}, \quad (8)$$

$$\dot{\mathbf{u}}_{t_k} \in [\dot{\mathbf{u}}_{\min}, \dot{\mathbf{u}}_{\max}], \forall t_k \in \mathbb{N}_{[k, k+N-1]}, \quad (9)$$

$$\mathbf{p}_{t_k} \notin \mathcal{O}, \forall t_k \in \mathbb{N}_{[k, k+N-1]}, \quad (10)$$

$$\mathbf{p}_{t_k} \notin \mathcal{D}_{t_k}, \forall t_k \in \mathbb{N}_{[k, k+N_{\text{crit}}]}. \quad (11)$$

where n_r is the number of robots, N_{crit} is the critical horizon used to apply hard constraints on avoiding dynamic objects, $\dot{\mathbf{u}}_{t_k}$ is the derivative of the action, and $(\mathbf{u}_{\min/\max}, \dot{\mathbf{u}}_{\min/\max})$ are physical limits of the robot's action.

V. MULTI-STEP MULTIMODAL MOTION PREDICTION

This section introduces a key innovation: applying energy-based learning to discrete maps for multimodal motion prediction, framed as a large-scale classification problem. It is shown that the pixel-level Negative Log-Likelihood (NLL) loss can produce concentrated predictions required for downstream planning. The main challenge with training stability is addressed by modifying the NLL loss.

A. Energy-based Training

Energy-based models [26] are models trained to yield energy spaces for given inputs. The training process is known as EBL. While commonly used in generative models, they can be applied to regression tasks [27]. For a model $G_\theta(\cdot)$ parameterized by θ , given input \mathbf{x} and label \mathbf{y} , it should assign a low energy to the label \mathbf{y} and higher energy for others \mathbf{y}' , i.e., $G_\theta(\mathbf{x}, \mathbf{y}) \leq G_\theta(\mathbf{x}, \mathbf{y}')$. In motion prediction, given input such as historical pedestrian motion, the model generates an energy space $\mathbf{E} = G_\theta(\mathbf{x})$ for future motion. The predicted future position $\hat{\mathbf{y}}$ is obtained by querying the point in \mathbf{E} with the lowest energy. The desired energy space should be constructed to minimize $\|\mathbf{y} - \hat{\mathbf{y}}\|$. Treating \mathbf{E} as an unnormalized probability map, multimodal motion predictions can be obtained via Monte Carlo sampling.

According to the Gibbs-Boltzmann distribution [26], the energy can be converted into probability densities,

$$P_c := \Pr(\mathbf{y}|\mathbf{x}; \theta) = \frac{e^{-G_\theta(\mathbf{x}, \mathbf{y})}}{\int e^{-G_\theta(\mathbf{x}, \mathbf{y}')} d\mathbf{y}'}. \quad (12)$$

The integral is mostly intractable [28] and needs to be estimated, e.g., via importance sampling [27]. It is possible to discretize the energy space and substitute the integral by summation. For motion prediction, if the discretized grid size is equal to the resolution of the environmental image, the approximation of the integral is sufficiently accurate. In the two-dimensional situation, for a discrete energy map with width W and height H , Eq. (12) can be rewritten as:

$$P := \Pr(\mathbf{y}|\mathbf{x}; \theta) = \frac{e^{-\mathbf{E}_{w,h}}}{\sum_{w'=1}^W \sum_{h'=1}^H e^{-\mathbf{E}_{w',h'}}}. \quad (13)$$

where $\mathbf{y} = [w, h]^T$ indicating the row index w and column index h on the image axis. Even though Eq. (13) seems reasonable, training accordingly is unstable, and the result tends to be noisy when the size of the energy surface is large. For example, given an energy map of size 100×100 , for each (\mathbf{x}, \mathbf{y}) , only the energy of one pixel is pushed down and the energy of the other 9999 pixels is pulled up. This makes the model sensitive to noise and likely to generate too high or low energy. To solve this problem, the weighted soft loss and a modified exponential layer [29], called the Positive Exponential Linear Unit (PELU) are proposed.

The weighted soft loss utilizes relaxed label masks. Instead of a single ground-truth pixel \mathbf{y} , a Gaussian distribution mask \mathbf{A} centered at \mathbf{y} with the same size of \mathbf{E} is used. Supposing the constructed mask \mathbf{A} to be the ground-truth distribution, which typically cannot be validated in the real world, the model can be trained to approximate this distribution via the Binary Cross Entropy (BCE) loss [18] or Kullback–Leibler Divergence (KLD). If the model is trained via BCE (which is not EBL), the direct output logits need to be processed by a Sigmoid layer and normalized into probability maps. Let $\hat{\mathbf{A}}$ be the normalized mask (\mathbf{A} divided by its maximum value).

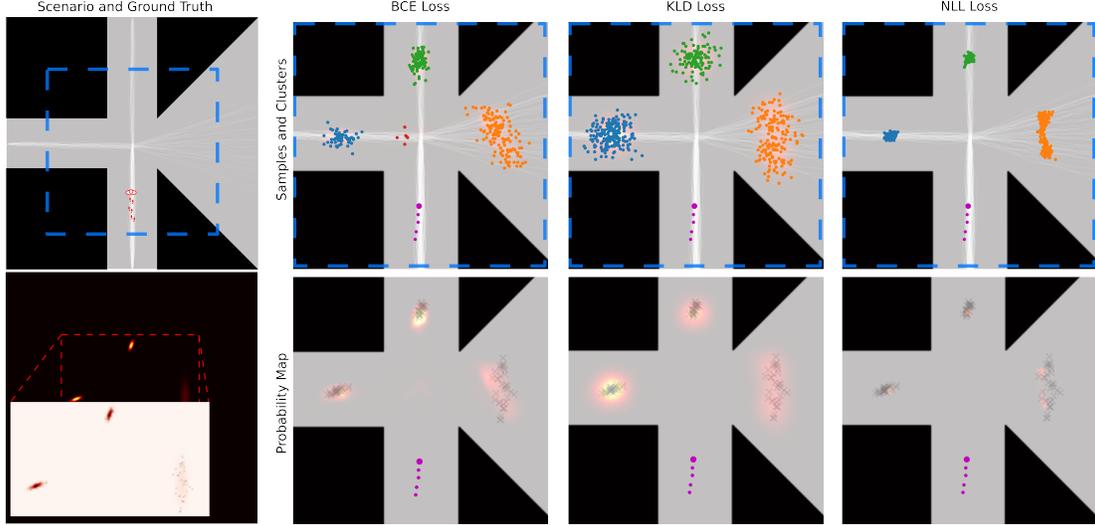


Fig. 3. Comparison of training results on the synthetic crossroad dataset from different learning strategies. The first column shows the scenario (top) with simulated trajectories (white lines) and the estimated ground-truth probability distribution (bottom) after 20 time steps. For the other subfigures, purple dots indicate the current (big) and past (small) positions of the target. The first row contains predicted samples and clusters from different methods, and each color indicates a cluster. In the second row, black crosses on probability maps are the ground-truth future positions.

The BCE loss function can be written as

$$\begin{aligned} \mathcal{L}_{BCE} = & - \sum_{w',h'} [\bar{\mathbf{A}}_{w',h'} \ln \sigma(\mathbf{E}_{w',h'})] \\ & - \sum_{w',h'} [(1 - \bar{\mathbf{A}}_{w',h'}) \ln(1 - \sigma(\mathbf{E}_{w',h'}))]. \end{aligned} \quad (14)$$

where $\sigma(\cdot)$ is the element-wise Sigmoid function. Note that the notation \mathbf{E} is kept for simplicity, but the output is not an energy space for BCE-trained models. If the model is trained via KLD (which is one form of EBL), the loss function is

$$\begin{aligned} \mathcal{L}_{KLD} = & \sum_{w',h'} \mathbf{A} \ln \frac{\mathbf{A}}{\mathbf{P}} = \sum_{w',h'} \mathbf{A}_{w',h'} \mathbf{E}_{w',h'} \\ & + \sum_{w',h'} \mathbf{A}_{w',h'} \ln \sum_{w',h'} e^{-\mathbf{E}_{w',h'}}. \end{aligned} \quad (15)$$

Both BCE and KLD regard the constructed Gaussian mask as the ground truth and attempt to align the predicted distribution with it, which can lead to overestimation, as shown in Fig. 2. This overestimation increases the likelihood of covering the actual trajectory, particularly in open areas where dynamic objects exhibit more random movement compared to constrained areas such as factories. However, it complicates downstream motion planning by covering excessively large areas, potentially obstructing the entire drive space for AMRs.

To overcome this issue, rather than using \mathbf{A} as a distribution, its normalized mask $\bar{\mathbf{A}}$ is applied as a weight mask for the energy space to redefine the probability $\bar{\mathbf{P}}$,

$$\bar{\mathbf{P}}_{w,h} = \frac{e^{-\bar{\mathbf{A}}_{w,h} \mathbf{E}_{w,h}}}{\sum_{w'=1}^W \sum_{h'=1}^H e^{-\mathbf{E}_{w',h'}}}, \quad (16)$$

and the NLL loss is applied,

$$\begin{aligned} \mathcal{L}_{NLL} = & - \ln \bar{\mathbf{P}} = - \ln \sum_{w',h'} \bar{\mathbf{A}}_{w',h'} e^{-\mathbf{E}_{w',h'}} \\ & + \ln \sum_{w',h'} e^{-\mathbf{E}_{w',h'}}. \end{aligned} \quad (17)$$

From our training results, as shown in Fig. 2, NLL generates more concentrated results compared to BCE and KLD.

As an EBL method, the NLL loss aggressively shapes the energy space, pushing the energy at the ground-truth cells down to a large negative value. A large negative energy leads e^{-E} to infinity and thus fails the training. Our solution is to add an output layer for energy regulation by substituting the exploding part of exponentiation with a linear function, which is similar to the ELU activation function [29],

$$f_{\text{PELU}}(x) = \max(0, x) + \min(0, e^x - 1) + 1 + \epsilon, \quad (18)$$

where ϵ is a small positive offset to prevent the output from reaching zero, ensuring the logarithm remains finite. Since the output layer only produces positive values, it is called Positive ELU (PELU). The PELU layer generates a processed energy map $\mathbf{E}' = \text{PELU}(-\mathbf{E})$, which can be converted into probability maps \mathbf{P} according to Eq. (13),

$$\mathbf{P}_{w,h} = \frac{\mathbf{E}'_{w,h}}{\sum_{w'=1}^W \sum_{h'=1}^H \mathbf{E}'_{w',h'}}. \quad (19)$$

By combining Eq. (17) and (19), the final Energy-oriented NLL (ENLL) loss function is

$$\mathcal{L}_{\text{ENLL}} = - \ln \sum_{w',h'} \bar{\mathbf{A}}_{w',h'} \mathbf{E}'_{w',h'} + \ln \sum_{w',h'} \mathbf{E}'_{w',h'}. \quad (20)$$

B. Clustering and Gaussian Fitting with Probabilities

Samples can be drawn at each time offset on the probability map. By clustering these samples, different modes are represented by distinct clusters. Gaussian fitting is then applied to the samples in each cluster, resulting in multimodal elliptical Gaussian estimations of the future position. Alternatively, samples can be directly fit into a Gaussian Mixture Model (GMM). This process, called Clustering and Gaussian Fitting (CGF) [23], yields the estimated future states $\{\{\hat{\mathbf{z}}_{t_k}^{(i)}\}_{m=1}^M\}_{t_k=k}^{k+N}$ of dynamic obstacle areas $\{\hat{\mathcal{D}}_{t_k}\}_{t_k=k}^{k+N}$. Since each sample

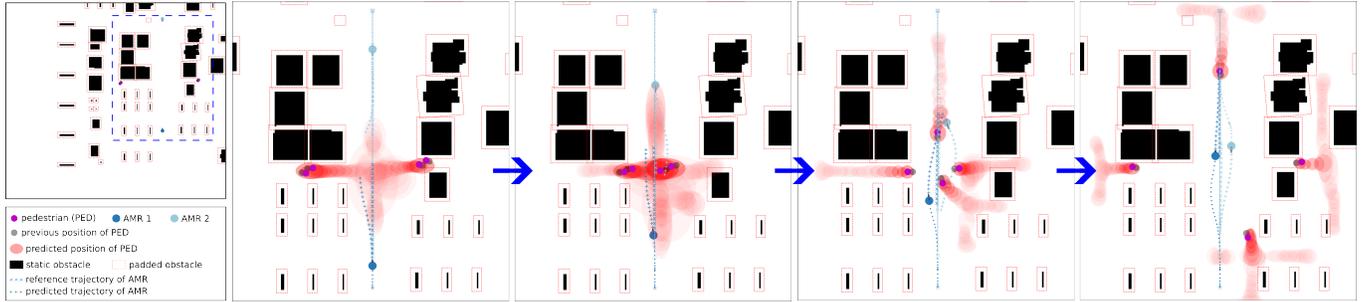


Fig. 4. Scenario 3 in obstacle avoidance evaluation. Four pedestrians and two AMRs are crossing an intersection, which causes a busy area and is challenging for AMRs. The illustrated collision-free example is from MPC-ENLL. The subfigures are in chronological order from left to right.

TABLE I
QUANTITATIVE COMPARISON OF DIFFERENT METHODS IN FIG. 3.

| Metrics (lower is better) | BCE | KLD | NLL |
|---------------------------|--------|---------------|--------------|
| JSD | 0.4036 | 0.3407 | 0.3482 |
| SWD ($\times 10^{-5}$) | 9.843 | 9.958 | 7.407 |

has a probability density, the probability of a cluster can be estimated by either the maximum or the weighted average probability of the samples in that cluster. If a GMM is directly obtained, the component weights can serve as the cluster probabilities.

VI. IMPLEMENTATION

A. Motion Prediction via Deep Learning

A U-Net architecture [30] with the PELU output layer is utilized to generate one-shot motion predictions. The network consists of four downsampling layers and four corresponding upsampling layers with skip connections. A lite version with 32/64/128/256 filters is implemented. More detailed hyperparameters can be found in the configuration from the provided repository. The input \mathbf{x} to the neural network is a stack of masks I_m , indicating the targeted object's positions, and a scene image I_s . The input can be regarded as a multi-channel image and each channel has the same size $W \times H$. The output is a stack of processed energy spaces, and each of them indicates the future position distribution at a specific time instant. The variance of the Gaussian distribution mask \mathbf{A} is 10 pixels. The model is trained on four datasets. The first one is the Stanford Drone Dataset [25] as shown in Fig. 2 for qualitative comparison, and the training process follows the same procedure as in [18]. The second one is a synthetic crossroad dataset for quantitative analysis, with 300 trajectories generated for training, as shown in Fig. 3. The other ones include a warehouse dataset [11] and a hospital dataset for the validation of the proposed DOV pipeline. The warehouse synthetic dataset contains 580 trajectories that are collected in the environment with random motion noise and the background image resolution is 330×293 . The hospital synthetic dataset contains 660 trajectories with larger motion uncertainty than the warehouse scene and the background image resolution is 321×321 . In synthetic datasets, the pedestrian model is omnidirectional and moves along predefined paths with random velocity noise.

B. Proactive Collision Avoidance

As illustrated in Fig. 1, given environmental data and the locations of AMRs and dynamic obstacles from a vision and tracking system, the multimodal motion predictor outputs stacked probability maps indicating future positions of the target object at each time step from 1 to N . After drawing samples from these probability maps, the CGF post-processing method is applied for each time step, producing a stack of geometric maps, with each map containing predicted potential ellipsoid obstacles. MPC then uses these stacked geometric maps to plan a collision-free trajectory for the AMR. The motion model for AMRs is the non-holonomic unicycle model as in [11]. The sampling time of the system is 0.2 s. The predictive horizon N is 20, which means 4 s ahead. In MPC, the critical horizon N_{crit} is 5. For the DOA term, both the model weight α_m and the time weight β_m are pre-defined. The model weight is currently not used and set to be 1. The time weight is designed to decrease as the time step increases. Values of all weights can be found in the configuration file from the given code repository.

VII. EVALUATION

In this section, both qualitative and quantitative evaluations of the proposed method are introduced. Realistic simulation is presented via ROS 2 and Gazebo¹. Full video of different scenarios is available².

A. Motion Prediction Evaluation

In motion prediction tasks, the capture of diversity is emphasized to increase the likelihood of matching the ground-truth trajectory. For downstream planning, the focus is on ensuring the predictions are closely aligned with the ground-truth probability distribution. As implied in Fig. 2, training via NLL produces concentrated predictions. Since real-world datasets don't come with ground-truth distributions and it is difficult to estimate a distribution due to the lack of data, for quantitative analysis, a simulated crossroad scene is used as shown in Fig. 3, where a pedestrian may go straight, turn left or right at the intersection. To increase local uncertainty, the right side is an open area, and pedestrians randomly pick endpoints along the right edge of the map. The ground-truth distribution

¹Code repository: https://github.com/Woodenonez/DyObAv_MPCnEBM_Warehouse

²Full video: <https://youtu.be/DpeadFZgl-Y>

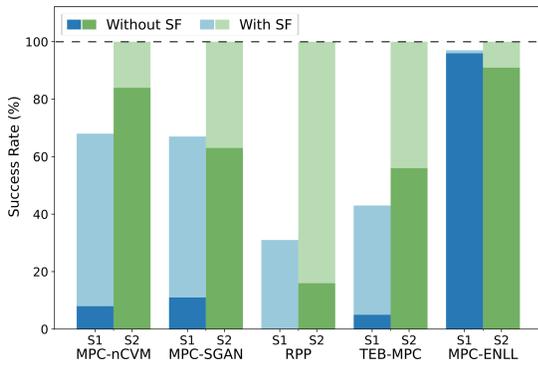


Fig. 5. Success rates of different methods with and without the cooperative SF pedestrian model. S1 and S2 mean Scenario 1 and Scenario 2.

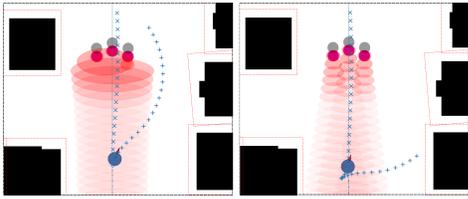


Fig. 6. Comparison between grouped prediction (left) and individual prediction (right). As in Fig. 4, the blue circle is the AMR, and the magenta circles are pedestrians. Red ellipses are predicted positions of pedestrians.

is estimated by fitting samples into a GMM (bottom left corner in Fig. 3) and compared to output probability maps from different models in terms of Jensen-Shannon Divergence (JSD) and Earth Mover’s Distance (EMD, or the Wasserstein distance [17]). JSD measures overall similarity rather than the actual distance between distribution peaks. EMD calculates the minimum “effort” to transform one distribution into another, which is the product of the amount of moved probability mass and the corresponding distance. EMD depends on both the distance between distributions and their overall shapes. Given EMD’s computational complexity in high dimensions, Sliced Wasserstein Distance (SWD) [31] is chosen for Monte Carlo approximation. As shown in TABLE I, KLD and NLL perform similarly on JSD, while NLL outperforms others on SWD. In our experiments, the original NLL loss performs adequately in the simple crossroad dataset. However, without the PELU and soft loss modifications, the training is unstable and prone to failure for real-world and complex datasets. The model also generates severe noise without the PELU layer, making it infeasible for downstream tasks.

B. Obstacle Avoidance Evaluation

The obstacle avoidance performance is evaluated by ensuring that AMRs avoid collisions with all obstacles. To thoroughly assess the proposed MPC-ENLL method, we compare various obstacle avoidance strategies, with and without cooperative pedestrian behavior, and then examine the impact of different predictors, including BCE (non-EBL), KLD (EBL), and ENLL (EBL). The effectiveness of the CGF grouping is also validated. It is important to note that in all tests, motion noise is added to human movement.

Four test scenarios are set. Scenarios 1-3 are in a warehouse and Scenario 4 is in a hospital area, as shown in Fig. 7.

Scenarios 1 and 2 are inherited from [11]. In Scenario 1, a pedestrian suddenly emerges from a corner, while in Scenario 2, a pedestrian abruptly changes direction in front of a robot. Scenario 3 involves two AMRs and four pedestrians meeting at an intersection, as in Fig. 4. In Scenario 4, a longer test is conducted, where an AMR operates in a hospital area and interacts with two pedestrians multiple times. The complete episode is available in the attached video. Multiple metrics are adopted, including linear and angular action smoothness (second derivative), clearance to obstacles, deviation from the reference path, solving time of motion planner (the duration from the receipt of motion prediction results and reference trajectories to the generation of the action), and success rate (defined as reaching the goal within a time limit without any collisions). In addition to MPC-WTA and DWA-CVM methods [11], several baselines are evaluated: Regulated Pure Pursuit (RPP) [32] which is a stop-and-wait method, Timed-Elastic Band (TEB) [33] which is a trajectory optimization method that generates collision-free reference trajectories for MPC, CVM with angular noise (nCVM) [12] integrated with MPC, and learning-based Social-GAN (SGAN) [16] with MPC. The results are in TABLE II. While learning-based control methods are not in the evaluation, as this study focuses on deterministic and optimization-based approaches, according to the study in [22], learning-based methods demonstrate superior performance in avoiding complex static obstacles but struggle with moving obstacles with limited training. The use of learning-based techniques is a potential future extension of this work. As in TABLE II, our MPC-ENLL outperforms the other methods on success rate. Methods using the nCVM or SGAN predictor exhibit low success rates in Scenario 1 due to their failure to capture multimodal motion. Clearance is also notable. With proper motion prediction, MPC-ENLL maintains a large clearance from dynamic obstacles while utilizing space around static obstacles. For motion prediction, on an NVIDIA GTX 1650 Max-Q, the model inference time of the one-shot EBL predictor is about 3 ms per object, which is about 19 times faster than the WTA-based predictor (58 ms).

In TABLE II, all pedestrians in Scenarios 1 and 2 are assumed to be non-cooperative and ignore AMRs, representing a worst-case scenario. To evaluate a more common condition with cooperative pedestrians and test the robustness, a modified Social Force (SF) model [13] is applied, where pedestrians avoid AMRs actively. As shown in Fig. 5, in Scenario 1, due to the pedestrian’s abrupt appearance, even with the SF pedestrian model, other methods still struggle. In Scenario 2, all methods achieve perfect collision avoidance. These results highlight the difference in the collision-avoidance ability of different methods across scenarios, with the MPC-ENLL method demonstrating prominent performance regarding cooperative or non-cooperative dynamic obstacles. To compare different loss functions in network training, MPC-BCE and MPC-KLD are also tested as in TABLE II. In Scenarios 1, 3, and 4, they have lower success rates compared to the proposed MPC-ENLL method due to the overestimation of future occupied areas. In Scenario 2, with larger free space than the other scenarios, they achieve more comparable performance.

The FRP is considered when selecting the CGF grouping

TABLE II
EVALUATION RESULTS OF DIFFERENT METHODS (AVERAGE OVER 100 RUNS)

| Scenario | Method | Smoothness | | Clearance (m) | | Deviation (m) | | | Solving time (sec) | | Success (%) |
|------------|-----------------|--------------|--------------|---------------|---------|---------------|-------|-------|--------------------|--------------|-------------|
| | | linear | angular | static | dynamic | mean | std | max | mean | max | |
| Scenario 1 | MPC-WTA* [11] | 0.030 | 0.039 | 0.501 | 0.786 | 0.434 | 0.267 | 1.095 | 0.050 | 0.125 | 94 |
| | DWA-CVM* [34] | 0.045 | 0.094 | 0.989 | 0.308 | 0.204 | 0.142 | 0.704 | 0.195 | 0.413 | 48 |
| | MPC-BCE | 0.048 | 0.050 | 0.218 | 0.612 | 0.498 | 0.301 | 0.99 | 0.012 | 0.103 | 57 |
| | MPC-KLD | 0.046 | 0.054 | 0.254 | 0.515 | 0.475 | 0.289 | 0.999 | 0.010 | 0.102 | 43 |
| | MPC-ENLL | 0.040 | 0.048 | 0.333 | 0.704 | 0.632 | 0.375 | 1.270 | 0.008 | 0.036 | 96 |
| Scenario 2 | MPC-WTA* | 0.046 | 0.042 | 0.574 | 0.863 | 0.224 | 0.160 | 1.064 | 0.052 | 0.121 | 81 |
| | DWA-CVM* | 0.042 | 0.034 | 0.651 | 0.298 | 0.205 | 0.109 | 0.398 | 0.207 | 0.323 | 62 |
| | RPP [32] | 0.027 | ≈ 0 | 0.5 | 0.393 | 0.191 | 0.109 | 0.3 | ≈ 0 | ≈ 0 | 16 |
| | TEB-MPC [33] | 0.010 | 0.012 | 0.395 | 0.474 | 0.206 | 0.111 | 0.453 | 0.005 | 0.102 | 56 |
| | MPC-nCVM [12] | 0.046 | 0.031 | 0.475 | 0.730 | 0.229 | 0.115 | 0.500 | 0.011 | 0.103 | 84 |
| | MPC-SGAN [9] | 0.031 | 0.019 | 0.472 | 0.553 | 0.243 | 0.269 | 2.893 | 0.007 | 0.102 | 63 |
| | MPC-BCE | 0.049 | 0.048 | 0.303 | 1.601 | 0.391 | 0.149 | 0.859 | 0.010 | 0.103 | 89 |
| | MPC-KLD | 0.058 | 0.049 | 0.366 | 1.094 | 0.343 | 0.125 | 0.574 | 0.011 | 0.103 | 75 |
| | MPC-ENLL | 0.040 | 0.024 | 0.492 | 0.927 | 0.210 | 0.109 | 0.479 | 0.010 | 0.103 | 91 |
| Scenario 3 | MPC-ENLL | 0.027 | 0.048 | 0.733 | 0.898 | 1.262 | 0.941 | 2.201 | 0.013 | 0.103 | 77 |
| Scenario 4 | MPC-BCE | 0.027 | 0.033 | 0.042 | 0.731 | 0.585 | 0.397 | 1.226 | 0.037 | 0.121 | 19 |
| | MPC-KLD | 0.029 | 0.030 | 0.071 | 0.662 | 0.918 | 0.708 | 2.401 | 0.027 | 0.118 | 18 |
| | MPC-ENLL | 0.017 | 0.022 | 0.099 | 0.586 | 0.017 | 0.350 | 0.893 | 0.019 | 0.109 | 90 |

Methods in bold font are proposed in this work. Metrics in bold font are the best results. Starred (*) results are from [11]. In all scenarios, results of methods with success rates lower than 15% are omitted. Scenario 3 shows worse data in two robots.

strategy. As illustrated in Fig. 6, where three pedestrians are working towards an AMR, if the individual prediction is implemented, these predictions form a non-convex obstruction causing the robot to freeze. After grouping predictions via CGF grouping, the AMR regards the occupied area in its entirety and successfully avoids it.

In all MPC experiments, the solver time is restricted to 0.1 s for real-time performance, meaning that the solving process can be interrupted. This can lead to premature solutions without the collision-free guarantee [11]. In production, a dedicated processor typically runs the solver that is further optimized. Additional safety layers including extra sensors are implemented to ensure safety. These measures are beyond the scope of this work and are not discussed.

C. Long-term Robustness

Apart from Scenario 4, to further examine the robustness of the proposed pipeline, long-term running experiments in the warehouse (with cooperative SF pedestrians) are implemented. To increase the uncertainty and interaction frequency, three AMRs and three pedestrians operating in close proximity are included, set to follow predefined routes. The location and timing of interactions between AMRs and pedestrians varied randomly due to the human motion noise. In three trials, each AMR encountered over 20 pedestrian interactions (defined as instances where a pedestrian influenced the robot's actions) and multiple fleet collision avoidance cases. The proposed approach demonstrates robustness and adaptability throughout the tests. Robots autonomously select different strategies based on the situation, such as taking detours when necessary, cutting shorter paths if safe, reversing in response to dynamic obstacles, etc. With distributed MPC, collision avoidance between multiple AMRs is also handled smoothly.

D. Simulation in Gazebo

The proposed pipeline is deployed in Gazebo for realistic simulation. Fig. 7 demonstrates the warehouse and hospital

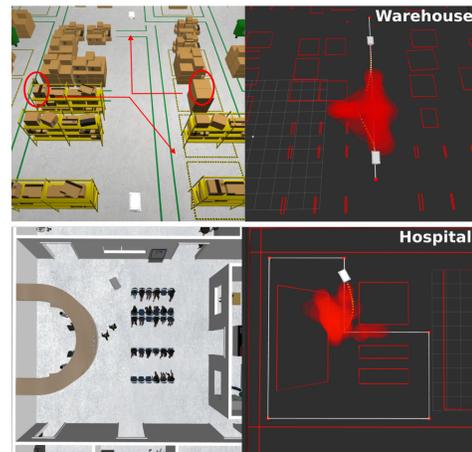


Fig. 7. Gazebo simulation of Scenarios 3 (up) and 4 (down). Red circles and arrows indicate the initial positions and intended routes of pedestrians. In RViz visualization, red ellipses are predictions from the energy-based predictor and yellow dotted lines are planned trajectories by MPC.

environments. The corresponding code and environments are available in the provided repository.

E. Relevant Hyperparameters and Normalized Masks

In this session, we briefly discuss some relevant hyperparameters and analyze the use of normalized masks. In the original Gibbs-Boltzmann distribution, there is an inverse temperature parameter scaling energy values in the numerator of (12), which is normally 1 as in this work. In practice, this can be used to tune the conservativeness of the prediction. Another parameter that affects the conservativeness of outputs is the variance of the ground-truth Gaussian mask \mathbf{A} . BCE and KLD are more sensitive to the variance of the mask, as they attempt to match it exactly. The proposed ENLL loss is less affected by the mask width, offering greater robustness.

In different loss functions, different forms of masks are employed, either the estimated ground-truth probability map \mathbf{A} or its normalization $\hat{\mathbf{A}}$. For the BCE loss, normalization

is necessary for mathematical soundness since it requires a Sigmoid output layer. Otherwise, it becomes inapplicable. For the KLD loss, as defined in (15), normalizing the mask scales it by the reciprocal of the maximum value in \mathbf{A} . When evaluating using the setup in TABLE I, with normalized masks, the JSD is 0.3267 and the SWD is 9.6767×10^{-5} , which are slightly better than KLD without normalized masks but not significantly. Moreover, normalization had no essential impact on obstacle avoidance performance as it cannot solve the problem of over-conservativeness. For the ENLL loss, using the normalized mask scales up its negative component, which may bias training toward stronger alignment with the mask. In practice, ENLL trained with unnormalized masks produced slightly more concentrated predictions but did not lead to significant differences in obstacle avoidance performance.

VIII. CONCLUSION AND FUTURE WORK

In this study, we proposed a new energy-based learning method for multimodal motion prediction and extended the MPC formulation for fleet coordination in an integrated collision-free navigation pipeline for autonomous mobile robots. The motion predictor based on energy-based learning is shown to be more suitable for downstream motion planning tasks. The combined MPC-ENLL approach outperforms other popular obstacle avoidance approaches with or without the cooperative pedestrian model. This pipeline is also implemented in ROS2 and simulated in Gazebo under a warehouse environment, which shows feasibility and safety.

One limitation of this work is the use of simulated environments, with complex static–dynamic obstacle interactions left for future investigation. Other future work could include three aspects: extending the pipeline to include the vision and tracking system; implementing a reference generator for MPC to achieve fast convergence; and considering industrial transport tasks involving more mobile robots and task schedulers.

REFERENCES

- [1] M. De Ryck, M. Versteyhe, and F. Debrouwere, “Automated guided vehicle systems, state-of-the-art control algorithms and techniques,” *Journal of Manufacturing Systems*, vol. 54, pp. 152–173, 2020.
- [2] P. Trautman and A. Krause, “Unfreezing the robot: Navigation in dense, interacting crowds,” in *IROS*, 2010, pp. 797–803.
- [3] A. J. Sathyamoorthy, U. Patel, T. Guan, and D. Manocha, “Frozone: Freezing-free, pedestrian-friendly navigation in human crowds,” *Robotics and Automation Letters*, vol. 5, no. 3, pp. 4352–4359, 2020.
- [4] S. Liu, P. Chang, Z. Huang, N. Chakraborty, K. Hong, W. Liang, D. L. McPherson, J. Geng, and K. Driggs-Campbell, “Intention aware robot crowd navigation with attention-based interaction graph,” in *ICRA*, 2023, pp. 12 015–12 021.
- [5] C. Mavrogiannis, F. Baldini, A. Wang, D. Zhao, P. Trautman, A. Steinfeld, and J. Oh, “Core challenges of social robot navigation: A survey,” *ACM Transactions on Human-Robot Interaction*, vol. 12, no. 3, 2023.
- [6] C. Cao, P. Trautman, and S. Iba, “Dynamic channel: A planning framework for crowd navigation,” in *ICRA*, 2019, pp. 5551–5557.
- [7] L. Huber, J.-J. Slotine, and A. Billard, “Avoiding dense and dynamic obstacles in enclosed spaces: Application to moving in crowds,” *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 3113–3132, 2022.
- [8] N. E. Du Toit and J. W. Burdick, “Robot motion planning in dynamic, uncertain environments,” *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 101–115, 2012.
- [9] L. Heuer, L. Palmieri, A. Rudenko, A. Mannucci, M. Magnusson, and K. O. Arras, “Proactive model predictive control with multi-modal human motion prediction in cluttered dynamic environments,” in *IROS*, 2023, pp. 229–236.
- [10] A. Wang, C. Mavrogiannis, and A. Steinfeld, “Group-based motion prediction for navigation in crowded environments,” in *Conference on Robot Learning (CoRL)*, 2022, pp. 871–882.
- [11] Z. Zhang, H. Hajieghrary, E. Dean, and K. Åkesson, “Prescient collision-free navigation of mobile robots with iterative multimodal motion prediction of dynamic obstacles,” *Robotics and Automation Letters*, vol. 8, no. 9, pp. 5488–5495, 2023.
- [12] C. Schöller, V. Aravantinos, F. Lay, and A. Knoll, “What the constant velocity model can teach us about pedestrian motion prediction,” *Robotics and Automation Letters*, vol. 5, no. 2, pp. 1696–1703, 2020.
- [13] D. Helbing and P. Molnar, “Social force model for pedestrian dynamics,” *Physical review E*, vol. 51, no. 5, p. 4282, 1995.
- [14] J. van den Berg, M. Lin, and D. Manocha, “Reciprocal velocity obstacles for real-time multi-agent navigation,” in *ICRA*, 2008, pp. 1928–1935.
- [15] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, “Social LSTM: Human trajectory prediction in crowded spaces,” in *CVPR*, June 2016.
- [16] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, “Social GAN: Socially acceptable trajectories with generative adversarial networks,” in *CVPR*, June 2018.
- [17] O. Makansi, E. Ilg, O. Cicek, and T. Brox, “Overcoming limitations of mixture density networks: A sampling and fitting framework for multimodal future prediction,” in *CVPR*, 2019.
- [18] K. Mangalam, Y. An, H. Girase, and J. Malik, “From goals, waypoints & paths to long term human trajectory forecasting,” in *ICCV*, 2021, pp. 15 213–15 222.
- [19] D. Ridel, N. Deo, D. Wolf, and M. Trivedi, “Scene compliant trajectory forecast with agent-centric spatio-temporal grids,” *Robotics and Automation Letters*, vol. 5, no. 2, pp. 2816–2823, 2020.
- [20] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, “Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data,” in *European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 683–700.
- [21] K. Guo, W. Liu, and J. Pan, “End-to-end trajectory distribution prediction based on occupancy grid maps,” in *CVPR*, 2022, pp. 2232–2241.
- [22] K. Ceder, Z. Zhang, A. Burman, I. Kuangaliyev, K. Mattsson, G. Nyman, A. Petersén, L. Wisell, and K. Åkesson, “Bird’s-eye-view trajectory planning of multiple robots using continuous deep reinforcement learning and model predictive control,” in *IROS*, 2024, pp. 8002–8008.
- [23] Z. Zhang, E. Dean, Y. Karayiannidis, and K. Åkesson, “Motion prediction based on multiple futures for dynamic obstacle avoidance of mobile robots,” in *CASE*, 2021, pp. 475–481.
- [24] F. Bertilsson, M. Gordon, J. Hansson, D. Möller, D. Söderberg, Z. Zhang, and K. Åkesson, “Centralized versus distributed nonlinear model predictive control for online robot fleet trajectory planning,” in *CASE*, 2022, pp. 701–706.
- [25] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, “Learning social etiquette: Human trajectory understanding in crowded scenes,” in *European Conference on Computer Vision (ECCV)*. Springer, 2016.
- [26] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, F. Huang, *et al.*, “A tutorial on energy-based learning,” *Predicting structured data*, vol. 1, no. 0, 2006.
- [27] 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)*. Springer, 2020, pp. 325–343.
- [28] Y. Song and D. P. Kingma, “How to train your energy-based models,” *arXiv preprint arXiv:2101.03288*, 2021.
- [29] S. H. Djork-Arné Clevert, Thomas Unterthiner, “Fast and accurate deep network learning by exponential linear units (ELUs),” in *International Conference on Learning Representations (ICLR)*, 2016.
- [30] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2015, pp. 234–241.
- [31] N. Bonneel, J. Rabin, G. Peyré, and H. Pfister, “Sliced and radon wasserstein barycenters of measures,” *Journal of Mathematical Imaging and Vision*, vol. 51, pp. 22–45, 2015.
- [32] S. Macenski, S. Singh, F. Martín, and J. Ginés, “Regulated pure pursuit for robot path tracking,” *Autonomous Robots*, vol. 47, p. 685–694, 2023.
- [33] C. Rösmann, F. Hoffmann, and T. Bertram, “Timed-elastic-bands for time-optimal point-to-point nonlinear model predictive control,” in *ECC*, 2015, pp. 3352–3357.
- [34] M. Missura and M. Bennewitz, “Predictive collision avoidance for the dynamic window approach,” in *ICRA*, 2019, pp. 8620–8626.