Compositional Motion Generation From Demonstration With Object-Centric Neural Fields

Citation for the original published paper (version of record):
Tekden, A., Bekiroglu, Y. (2026). Compositional Motion Generation From Demonstration With Object-Centric Neural Fields. IEEE Robotics and Automation Letters, 11(9).
http://dx.doi.org/10.1109/LRA.2026.3713713

(article starts on next page)

Compositional Motion Generation from Demonstration with Object-Centric Neural Fields

Ahmet Tekden[1] Yasemin Bekiroglu[1,2]

***Abstract*—Compositionality, by organizing complex behavior as combinations of simpler elements, enables robot learning that is scalable and data efficient. Leveraging this principle, we propose a generative learning-from-demonstration framework that enables compositional modeling of robotic behavior by connecting perception and motion through shared object-level representations. We render scenes from object-centric neural representations that integrate canonical neural fields with latent-conditioned deformations, capturing positional and geometric variations in a smooth, consistent, and interpretable way. For motion generation, a temporal mixture-of-experts (MoE) employs a gating mechanism to combine object-conditioned movement primitives over time, producing complete trajectories. This spatial–temporal compositionality maintains the data efficiency of movement primitives while grounding motion in visual structure, enabling systematic generalization across diverse scene configurations. In simulation, long-horizon manipulation tasks are successfully completed using the proposed model, which requires significantly less training data than other image-based baselines. Real-world experiments further demonstrate the method's robustness to noise, its ability to generalize at the category level through language-based segmentation models, and its capacity to operate directly on 3D scene representations.**

***Index Terms*—Learning from Demonstration, Deep Learning in Grasping and Manipulation**

I. Introduction

Learning from Demonstration (LfD) [1] is a widely used approach for teaching robots task-specific skills from human demonstrations. A central challenge is representing diverse, long-horizon behaviors from a limited number of demonstrations. Treating motion as a single, monolithic sequence overlooks its compositional structure, which complicates the learning process. Everyday tasks unfold through sequential, object-level interactions—reaching, grasping, moving, and placing—that together accomplish the overall goal. This motion-level compositionality [2] can be complemented by scene-level compositionality, grounding motion in object-centric scene representations and providing a natural foundation for compositional motion modeling. Figure 1 illustrates an example of such object-centric structure, showing how the scene is decomposed into object-specific representations and how those representations relate to different phases of a manipulation task. By supporting data efficiency, modularity, and generalization, compositional formulations present a compelling alternative to monolithic approaches [3].

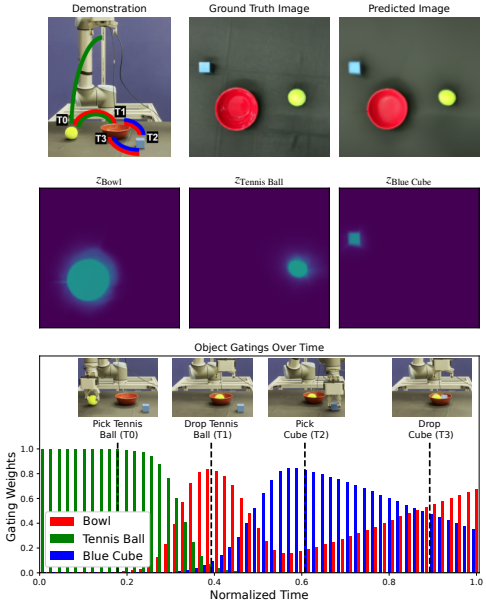

Fig. 1: System overview. Top: Task sequence with object interactions, followed by the ground-truth and reconstructed images. Middle: Object-specific masks inferred from latent codes, illustrating object-centric scene decomposition. Bottom: Temporal gating weights showing each object's influence over the trajectory; snapshots above the plot correspond to key events (T0–T3), marked by vertical dotted lines.

Movement primitives (MPs) [4]–[10] are widely used to encode and generalize demonstrated skills. MPs provide a compact and smooth representation of trajectories from only a few demonstrations by explicitly incorporating temporal variables into motion modeling. Typically, they are employed compositionally to represent atomic skills, which can then be combined to model more complex behaviors. However, this compositionality is often imposed manually, e.g., by concatenating primitives or specifying splits with via-points. Moreover, MPs generally rely on hand-designed, low-dimensional features (e.g., object positions, goal locations). Identifying suitable features and estimating them consistently across tasks is challenging. Alternatives include learning features directly from images [6], but this typically requires large datasets and sacrifices the data-efficiency advantage of MPs.

To address these limitations, we propose a generative approach to model scenes with smooth object-level parameterizations [1]. This enables task representations based on compact latent parameters that preserve the efficiency and compositionality of MPs while grounding them in perception.

Manuscript received: December 18, 2025; Revised May 25, 2026; Accepted June 24, 2026.

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

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation, the Chalmers AI Research Center (CHAIR), and the Chalmers Gender Initiative for Excellence (Genie). [1]Department of Electrical Engineering, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden. [2]Department of Computer Science, University College London, WC1E 6BT London, U.K. Email: tekden@chalmers.se

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

[1]Project page: https://fzaero.github.io/compositional/

Building on this perspective, we propose a Mixture-of-Experts (MoE)–based motion generation framework that integrates perception and action. A spatial MoE captures smooth, object-centric generative representations that provide the building blocks for motion modeling. Object-centric structure is encouraged by soft mask–based importance sampling, ensuring that each expert learns independently without collapsing onto the same object; because masks are used only for sampling, they do not need to align precisely with object boundaries. Trajectories are represented compositionally through temporal gating, where multiple experts can be active within a segment, reflecting the relational nature of manipulation (e.g., a transfer motion depending on both grasp and place poses). Together, this spatial–temporal MoE integrates perception and action, preserves the efficiency of primitive-based methods, and enables systematic generalization to novel scenes.

The contributions of this work are as follows. A scene modeling approach is proposed that learns smooth and interpretable object-centric representations, paired with a motion modeling approach that enables sequential compositionality through temporal gating across multiple experts. Building on these approaches, a data-efficient LfD framework is introduced that integrates compositional scene and motion modeling to learn movement primitives directly from visual input, supporting generalization from few demonstrations. The framework is validated in both simulation and real-world experiments, demonstrating improved efficiency and generalization compared with existing baselines.

II. Related Work

Movement primitives (MPs) are a dominant representation in LfD [1], providing smooth, low-dimensional encodings of trajectories. Extensions include probabilistic [5], [8], task-parameterized [6], [10], [11], and via-point–based formulations [7]. Despite their effectiveness, compositionality is usually imposed manually, and MPs typically rely on hand-crafted features. Neural field–based formulation has also been explored for scene- and motion-level modeling [9], but this approach operates at the scene level and scales poorly with the number of objects or scene parameters. Our framework addresses these limitations by modeling scenes and motions in a generative and compositional manner, thereby preserving the data efficiency of MPs. We further demonstrate category-level trajectory generation using object-level segmentation.

Recent imitation-learning and generative policy approaches, including diffusion-based and transformer-based methods [12], [13], have demonstrated strong performance in robotic manipulation. More recently, Vision-Language-Action (VLA) models and generalist robot policies have explored large-scale cross-task visuomotor learning using pretrained vision-language representations and large robot datasets [14], [15]. Several recent works further investigate object-centric and few-shot adaptation strategies for pretrained VLAs, such as ControlVLA [16], which combines online segmentation with pretrained visuomotor policies for object-conditioned manipulation. In contrast, our work focuses on few-demonstration, task-specific trajectory generation from a single static scene observation, where object-centric latent representations are inferred from the scene and used to condition full trajectory generation. Our experiments show that this formulation achieves high task performance given limited demonstrations.

Unsupervised scene decomposition methods, including Slot Attention [17] and neural field–based approaches such as GIRAFFE [18], learn object-centric scene representations from images. However, these methods typically rely on large-scale image datasets and are not designed for data-efficient robotic learning. In contrast, our framework learns object-centric representations from substantially fewer images and couples them with compositional motion modeling.

Neural fields [19] provide powerful continuous representations and have been extended beyond reconstruction [20]–[22] to robotics tasks. For motion generation, prior work has applied them to correspondence [23], [24], trajectory-level modeling [9] and reactive motion generation [25]. However, existing approaches either model motion only at the waypoint or pose level, focus on motion generation without supporting visual conditioning or sacrifice object-level compositionality and data efficiency. In contrast, we introduce a neural field–based framework for compositional scene and motion modeling at the object level, enabling efficient full-trajectory generation from visual input.

III. Problem Formulation

We focus on learning to generate motion from demonstrations, leveraging compositional representations of scenes and motion to enable generalization from a few examples. The training data consist of a set of images with coarse object masks and corresponding motion trajectories:

$$\mathcal{D} = \left\{ \left(I^{(k)}, \{M_i^{(k)}\}_{i=1}^{N}, \tau^{(k)} \right) \right\}_{k=1}^{K}, \tag{1}$$

where $I^{(k)}$ is the k-th image, $\{M_i^{(k)}\}_{i=1}^{N}$ are the object masks for N objects, and $\tau^{(k)} = \{q^{(k)}(t)\}_{t=1}^{T}$ is motion trajectory of length T, where $q^{(k)}(t)$ denotes the robot state at time t, such as the end-effector position. The dataset includes demonstrations that cover the boundary values of the task parameters (e.g., the minimum and maximum admissible object configurations). This ensures that the model observes the full range of object variability present in the task. Each object mask $M_i^{(k)}$ marks the image region of a single object. The masks should sufficiently cover the target object, but do not need to precisely align with object boundaries [2]. We assume that objects occupy distinct regions and that the background is consistent across demonstrations during training, including a fixed camera viewpoint and consistent scene layout. This makes annotation easier and ensures that image variations reflect object-level changes.

We assume trajectories are approximately time-aligned across demonstrations, so that each index t corresponds to the same task phase (e.g., approach, grasp, transport, release). Such trajectories can be obtained through motion-planned demonstrations or by aligning kinesthetic trajectories based on contact events or changes in motion profiles [11]. This establishes a shared temporal structure across demonstrations

[2]In practice, the masks are annotated using one or two rectangles.

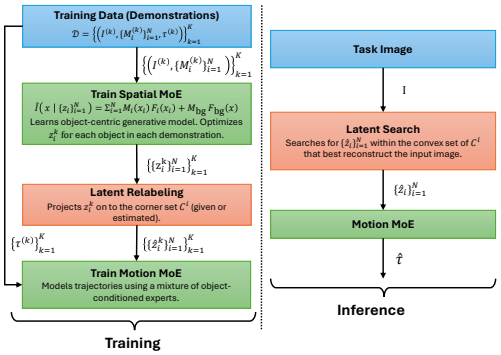

Fig. 2: Overview of training and inference pipelines. The Spatial MoE learns object-centric latent vectors, which are canonicalized via latent relabeling and used to train the Motion MoE. During inference, object latents recovered via latent search are used by the Motion MoE to generate trajectories.

and enables sequential compositionality, where different object latent representations influence distinct trajectory segments.

The learning objective is (i) to learn a scene representation model along with latent vectors $\{z_i\}_{i=1}^N$ for each object such that each scene can be expressed as a composition of a fixed background and objects whose variations (e.g., in position or geometry) are captured smoothly in the latent space; and (ii) to train a motion generation model that learns to map these latent vectors to the demonstrated trajectories. In the inference phase, latent vectors for a new scene are extracted using the learned scene representation model and fed into the motion model to generate the corresponding trajectory.

IV. Method

The proposed framework is shown in Fig. 2. We first learn an object-centric generative model from demonstration images (Sec. IV-A, Sec. IV-B), yielding latent representations that capture each object in the scene. We then construct a minimal canonical latent set that can equivalently reconstruct the training images (Sec. IV-C). Finally, we model motion compositionally by conditioning a temporal MoE on these canonical latent representations (Sec. IV-D), enabling trajectory generation grounded in object-centric structure.

A. Compositional Image Modeling

We model each image using an object-centric spatial MoE:

$$\hat{I}(x|\{z_i\}_{i=1}^N) = \sum_{i=1}^{N} M_i(x_i)\, F_i(x_i) + M_{\text{bg}}\, F_{\text{bg}}(x), \qquad (2)$$

with deformed coordinates $x_i = x - \Delta_i(x \mid z_i)$, where $F_i(x)$ predicts the appearance of object i, $\Delta_i(x \mid z_i)$ is the deformation conditioned on latent vector z_i, and $M_{\text{bg}}, F_{\text{bg}}$ denote the background mask and appearance model, respectively. All fields F_i, F_{bg}, and deformation Δ_i are implemented as coordinate-based multilayer perceptrons (MLPs), i.e., neural fields. The spatial MoE is shown in Figure 3.

Object masks are computed via a spatial softmax across objects and background:

$$M_i(x_i) = \frac{\exp(l_i(x_i))}{\exp(l_{\text{bg}}) + \sum_{j=1}^{N} \exp(l_j(x_j))}, \qquad (3)$$

where the background logit l_{bg} is constant, and each object logit is given by a learned function $l_i(x_i)$. The logit functions are also implemented as coordinate-based MLPs. This

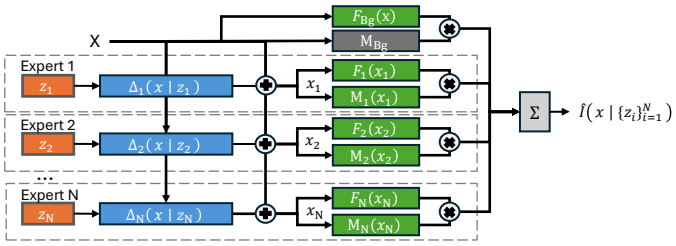

Fig. 3: Given N expert-specific objects, orange denotes learnable embeddings, blue corresponds to FiLM-conditioned MLPs, and green represents standard MLPs. Light gray indicates mathematical operators, and dark gray denotes the background mask computed from a constant, non-trainable logit.

formulation yields object masks that partition the image into spatially coherent regions, while deformations capture scene variations as smooth changes in object position and shape.

B. Object-Centered Training of Experts

Training the compositional image model proceeds in multiple stages. We first estimate the static background, then pretrain individual object fields using coarse masks, and finally refine all components jointly under the mixture formulation. Optimization in all stages uses an ℓ_2 reconstruction loss between predicted and ground-truth pixels. For deformation networks Δ_i, we additionally employ Lipschitz regularization [26] to encourage the mapping $z_i \mapsto \Delta_i(x \mid z_i)$ to vary smoothly with latent variables. While the neural-field parameterization already induces a generally smooth dependence on z_i, this regularization further helps by discouraging abrupt local changes and improving stability.

a) Background Modeling: We first train F_{bg} using only non-masked pixels from all demonstrated images. Compared to naïve median-image-based supervision, this procedure provides a more consistent background estimation.

b) Object-Centric Training: Each object model F_i is trained using its corresponding mask within an object-centric sampling framework. During training, pixels that do not belong to other objects are sampled from the demonstration image, with additional importance sampling applied to M_i to emphasize the appearance of the target object. Pixels that fall within the masks of other objects are instead sampled from the background model, preventing F_i from capturing regions associated with different objects. This sampling strategy ensures that variations across scenes are attributed solely to the correct object and enables the learned masks to refine coarse input masks into more precise segmentations.

To further improve training stability, we employ a position-aware zoom-based curriculum around each object mask. This approach is particularly beneficial for smaller objects, as it increases their effective pixel coverage and provides more stable gradients during optimization. The zoom factor is gradually reduced over time, allowing the model to converge toward accurate object representations at their true scale.

c) Joint Compositional Training: Finally, we continue training using the full compositional model in (2), with background and object models jointly optimized. The earlier stages act as pretraining, while this stage refines consistency between objects and background. Overall, this object-aware training scheme enables the model to explain variations across scenes

Input: Reference image I, corner set C^i for object i, number of samples P, preselection size L', maximum candidate count L, error threshold λ
Output: Top-L latent vector candidates $\{\hat{z}_{i,l}^{(k)}\}_{l=1}^{L}$

Sample P weight vectors $\{\alpha_p\}_{p=1}^{P} \sim \text{Dirichlet}(|C^i|)$
for $p = 1$ **to** P **do**
  $\hat{z}_{i,p}^{(k)} = \sum_j \alpha_{p,j}\, c_{i,j}, \quad c_{i,j} \in C^i$
  $E_{i,p}^{(k)} = \|I - \hat{I}(\cdot \mid \hat{z}_{i,p}^{(k)})\|_2$
end
Select the top-L' candidates with the lowest reconstruction errors
Refine these candidates via gradient descent on $E_{i,l}^{(k)}$
$E_{\min} = \min_l E_{i,l}^{(k)}$
Retain all candidates satisfying $E_{i,l}^{(k)} < \lambda E_{\min}$, up to L total
return $\{\hat{z}_{i,l}^{(k)}\}_{l=1}^{L}$

Algorithm 1: Latent Search

as object-dependent differences, allowing robust generalization with as few as 10–30 training images.

d) Extension to 3D Representations: While we describe F_i as 2D appearance fields, the mixture formulation is not restricted to images. Each F_i can also be defined as an occupancy field [22] over 3D space, enabling the construction of a compositional 3D scene representation.

C. Latent Relabeling

Each object is associated with a latent vector $z_i^{(k)} \in \mathbb{R}^{d_z}$ for each demonstration k. Smooth object variations are captured by interpolating between a small number of endpoint latent vectors that span the boundaries of the observed variation. For each object i, these endpoints form the corner set C^i. Given this set, the search (Algorithm 1) recovers object-specific latent representations for each demonstration by sampling convex combinations of the corners and refining the lowest-error candidates through gradient-based optimization. As multiple convex combinations can yield similar reconstructions, we retain the top-L candidates per demonstration to preserve valid alternatives for downstream motion modeling. This effectively increases the training set size from K to $K \times L$.

We estimate the corner sets by iteratively expanding C^i. We initialize C^i with two embeddings that are maximally distant within the set $\{z_i^{(k)}\}_{k=1}^{K}$ and use Algorithm 1 to test whether the remaining instances can be reconstructed from the current corner set. For any instance that cannot be reconstructed, we expand C^i by selecting the embedding most distant from the latents sampled via convex combinations of the current corner set, leveraging the observation that larger visual differences are generally reflected by greater distances in the latent space. This process iteratively widens the representable region of the latent space until all training instances can be reconstructed from the corner set, ensuring that every demonstration latent can be canonicalized as a convex combination of the corners. As a result, generalization is naturally constrained to variations represented within the demonstrated latent support, and extrapolation beyond this region is not guaranteed.

D. Motion Generation

Given the relabeled object latent vectors $\{\hat{z}_i\}_{i=1}^{N}$, we assume sequential compositionality: different trajectory segments depend on different subsets of $\hat{z}_i$. At an abstract level, we model this through a time-dependent conditioning signal constructed via softmax mixture weights. The exact conditioning mechanism is specified later in Sec. IV-E.

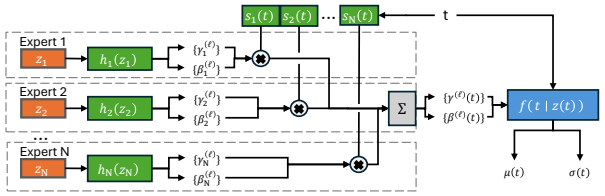

Fig. 4: Given N expert-specific object embeddings, FiLM weights and object relevance scores are predicted via MLPs (green). These are pooled to generate FiLM parameters used to condition the trajectory generation.

$$c(t) = \sum_{i=1}^{N} w_i(t)\, c_i(\hat{z}_i), \quad w_i(t) = \frac{\exp\big(s_i(t)\big)}{\sum_{j=1}^{N} \exp\big(s_j(t)\big)} \tag{4}$$

where $s_i(t)$ are learnable relevance scores and $c_i(\hat{z}_i)$ denotes object-specific conditioning variables. The generator then predicts a trajectory in the state space $q(t) \in \mathbb{R}^d$ as

$$q(t) \sim \mathcal{N}\big(\mu(t),\, \text{diag}\big(\sigma^2(t)\big)\big)\,. \tag{5}$$

where $\mu(t)$ and $\sigma(t)$ are estimated via

$$[\mu(t),\, \sigma(t)] = f(t \mid c(t)), \tag{6}$$

where $\sigma(t)$ is constrained to be positive via a softplus transform applied to the network output. The motion generation framework is shown in Figure 4.

The model parameters are optimized by minimizing the negative log-likelihood of the demonstrated trajectories, where each trajectory is modeled as a Gaussian:

$$\mathcal{L}_{\text{traj}} = -\sum_{k,t} \log \mathcal{N}\big(q^{(k)}(t)\,;\, \mu(t \mid c(t)),\, \Sigma(t \mid c(t))\big), \tag{7}$$

where $\Sigma(t \mid c(t)) = \text{diag}(\sigma^2(t \mid c(t)))$. To discourage the model from learning spurious dependencies on irrelevant latent variables, we apply latent dropout during early training by randomly masking a subset before pooling. This warm-up regularization encourages the generator to rely primarily on informative variables, while in later training and at test time, all variables are used. Additionally, the probabilistic trajectory formulation mitigates overfitting to spurious correlations by allowing irrelevant latent variables to be absorbed into the variance term rather than biasing the mean.

a) Modeling Orientation: For modeling orientation, we follow [27] and represent the quaternion trajectories $q_{1:T}$ in axis–angle form $r_{1:T}$, expressed in the tangent space of the initial quaternion q_1, yielding continuous rotation trajectories that can be modeled similarly to Cartesian trajectories.

E. Latent Conditioning with FiLM

Both the image model and the trajectory generator are conditioned on latent vectors. We implement this conditioning through *Feature-wise Linear Modulation (FiLM)* [28], which reconfigures the parameters of a base network using latent-dependent scale and shift terms. For a given z_i, FiLM produces parameters

$$h_i(z_i) = \{\gamma_i^{(\ell)},\, \beta_i^{(\ell)}\}_{\ell=1}^{L}, \tag{8}$$

where $\gamma_i^{(\ell)}, \beta_i^{(\ell)} \in \mathbb{R}^H$ for layer width H. Given base layer parameters $(w^{(\ell)}, b^{(\ell)})$, the modulated parameters are

$$\tilde{w}^{(\ell)} = \gamma^{(\ell)} \odot w^{(\ell)}, \quad \tilde{b}^{(\ell)} = \gamma^{(\ell)} \odot b^{(\ell)} + \beta^{(\ell)}, \tag{9}$$

with $\odot$ denoting elementwise multiplication. FiLM thus provides a lightweight mechanism to adapt a shared architecture to different objects or tasks. This role is analogous to hypernetworks [29], which generate full network weights from latent codes, but FiLM achieves similar conditioning capability with fewer parameters and improved stability.

In the compositional image model, FiLM parameters are derived from each z_i and applied directly when predicting object deformations. In the trajectory generator, the conditioning variables in (4) correspond to the FiLM modulation parameters, i.e., $c_i(\hat{z}_i) \equiv \{\gamma_i^{(\ell)}, \beta_i^{(\ell)}\}_{\ell=1}^{L}$. Accordingly, the aggregated conditioning signal is implemented through

$$\gamma^{(\ell)}(t) = \sum_{i=1}^{N} w_i(t)\, \gamma_i^{(\ell)}, \quad \beta^{(\ell)}(t) = \sum_{i=1}^{N} w_i(t)\, \beta_i^{(\ell)}. \tag{10}$$

This mechanism realizes sequential compositionality: different trajectory segments are influenced by different subsets of z_i, depending on which $w_i(t)$ are active.

F. Inference

At test time, we apply the same latent search procedure (Algorithm 1) using the corner sets from training. Objects are processed sequentially, with each contribution added incrementally to condition subsequent reconstructions. In rare cases where visually similar objects (e.g., of the same color) lead to ambiguity, larger objects are prioritized to provide more reliable reconstruction cues. Since reconstruction is based on visual similarity to the learned object representations, the method tolerates moderate lighting variation and visually distinct distractor objects. During latent inference, representations that better reconstruct the learned task-relevant object structure are favored, allowing distractor objects that are not well represented by the learned object model to be ignored. However, distractors that are visually similar to task-relevant objects may still lead to ambiguity or degraded performance. This procedure recovers object-specific latent representations for novel task instances while remaining within a compact and well-structured region of the latent space. The resulting latent representations then condition the trajectory generator. In the current implementation, running on an Intel(R) Xeon(R) W-2245 CPU @ 3.90GHz and an NVIDIA Quadro RTX 6000 GPU, inference takes approximately 1.4 and 0.8 seconds per object for RGB and 3D settings, respectively [3].

V. Experiments

We evaluate the framework through simulation benchmarks and real-world robotic experiments. The simulation studies show how compositional motion modeling through object-centric latent representations and time-varying latent aggregation improves generalization and data efficiency. Real-world experiments demonstrate the method's ability to handle perception noise, integrate 3D scene representations, and enable category-level generalization through vision-language–based segmentation.

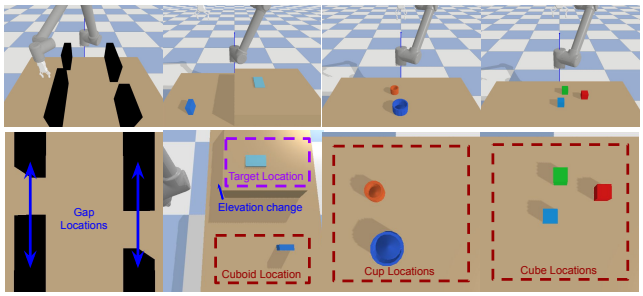

Fig. 5: Simulation tasks with corresponding scene variations. The bottom row shows the camera images used by the methods. In these images, arrows highlight geometric changes and rectangles mark possible object locations.

A. Simulation Experiments

We first assess the proposed method on four manipulation tasks of increasing complexity. In Wall Avoidance (easy), the robot reaches a goal by navigating through openings in two walls. Incline Pick-and-Place (medium) requires transporting a cuboid onto a platform at varying elevations, with both the cuboid and target locations changing across trials. Cup Stacking (medium) involves inserting a small cup into a larger one, while Cube Stacking (hard) requires stacking three cubes from diverse initial positions. These tasks introduce scene variations including geometric changes and varying object placements, as illustrated in Figure 5.

We compare the proposed method against several baselines, including conditional neural movement primitives (CNMP) [6] with CNN and Diffusion Policy (DP) [12]. CNMP supports image-conditioned trajectory generation, making it one of the closest comparable approaches for our setting. DP is a recent imitation-learning framework that has demonstrated strong performance in robotic manipulation [4]. To evaluate the proposed perception pipeline, we include CNN-FiLM and DINO-FiLM ablations that replace the object-centric latent representation with global image features. CNN-FiLM uses end-to-end learned CNN features, while DINO-FiLM uses pretrained DINO [30] features. We further include an ablation without temporal gating ("Ours w/o gating") to isolate the contribution of time-varying latent aggregation. Finally, we include a modified Neural Field Movement Primitive baseline (NFMP) [9], where we replace the hypernetwork with FiLM layers and omit positional encodings for a fairer comparison. Since NFMP relies on grid-based training over all combinations of task parameters, scaling as 3^n with n task parameters, it is evaluated only on the Wall Avoidance task, where the required number of demonstrations remains tractable.

Performance is evaluated using three criteria: mean Euclidean distance (MED) between predicted and demonstrated trajectories, task success rate, and data efficiency across varying training sizes. All models are trained with three random seeds, and results are reported as the mean and standard

[3] 3D inference employs sparsely sampled coordinates near surfaces instead of the full coordinate space.

[4] To align the prediction setting across methods, normalized time is provided as a state input to DP.

TABLE I: Simulation results (MED/Accuracy).

Method (data regime)	Wall Avoidance	Incline Pick-and-Place	Cup Stacking	Cube Stacking
CNN-CNMP (Low)	2.72±0.08/0.67±0.04	4.35±0.27/0.11±0.03	10.85±1.39/0.02±0.01	14.22±0.59/0.00±0.00
CNN-CNMP (High)	1.70±0.26/0.95±0.02	2.39±0.43/0.51±0.15	3.08±0.27/0.16±0.07	6.07±0.32/0.00±0.00
DP (Low)	2.31±0.04/0.79±0.02	3.60±0.08/0.25±0.01	11.13±1.29/0.02±0.02	11.55±0.19/0.00±0.00
DP (High)	0.55±0.03/**0.99±0.00**	1.49±0.14/0.86±0.04	2.72±0.09/0.26±0.04	4.77±0.61/0.11±0.03
NFMP (Grid)	0.85±0.18/0.98±0.01	NA	NA	NA
DINO-Film (Low)	4.46±0.19/0.65±0.02	5.03±0.13/0.21±0.00	9.81±0.34/0.00±0.00	15.10±0.07/0.00±0.00
DINO-Film (High)	2.27±0.12/0.84±0.00	3.71±0.06/0.31±0.06	6.69±0.06/0.01±0.00	10.48±0.08/0.00±0.00
CNN-Film (Low)	1.41±0.19/0.89±0.02	3.01±0.55/0.40±0.13	8.74±1.83/0.03±0.01	9.26±0.74/0.00±0.00
CNN-Film (High)	**0.46±0.09/0.99±0.00**	1.18±0.09/0.92±0.05	2.11±0.20/0.26±0.06	1.32±0.03/0.66±0.10
Ours w/o gating (Low)	0.71±0.04/**0.99±0.00**	0.89±0.03/0.99±0.01	1.10±0.03/0.86±0.05	1.33±0.05/0.85±0.06
Ours (Low)	**0.51±0.03/0.99±0.00**	**0.71±0.02/1.00±0.00**	**0.65±0.04/0.99±0.00**	**0.72±0.05/0.97±0.02**

Low/High training sizes per task: Wall Avoidance (10/30), Incline Pick-and-Place (20/50), Cup Stacking (20/50), and Cube Stacking (30/100). Bold numbers indicate the best performance in each task under the low- and all-data regimes, respectively. Lower MED and higher accuracy indicate better performance.

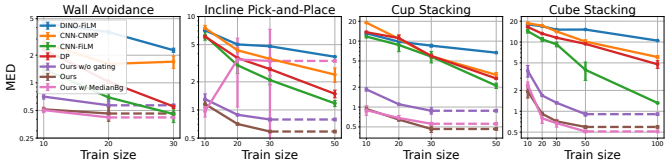

Fig. 6: Training size versus MED, illustrating the data efficiency of the proposed approach. The proposed method achieves low error with far fewer demonstrations than the baselines, particularly on more challenging tasks.

deviation across seeds. We consider two data regimes: a low-data regime with limited demonstrations per task, and a high-data regime with a larger demonstration set. Baselines are evaluated in both regimes, whereas the proposed method is evaluated only in the low-data regime, demonstrating that it matches or outperforms baselines trained with substantially more demonstrations.

Results are presented in Table I. Across tasks, the proposed method, both with and without object-centric temporal gating, consistently outperforms baselines in the low-data regime. The CNN-FiLM baseline generally achieves higher performance than CNN-CNMP and Diffusion Policy, suggesting that FiLM-conditioned trajectory generation itself provides a strong inductive bias in our setting. Even when trained with many more demonstrations, baseline performance is generally comparable to, or lower than, that of the proposed model. Comparing CNN-FiLM with the proposed method without temporal gating further highlights the contribution of the proposed object-centric representation. The only exception is the Wall Avoidance task, where CNN-FiLM achieves a marginally lower MED when trained with three times as many demonstrations. However, both methods successfully avoid the wall, and the relatively low task complexity reduces the data-efficiency advantage of the proposed method. To further illustrate data efficiency, Figure 6 shows MED as a function of training size, where each point represents the mean and standard deviation across seeds. The proposed method achieves consistently low error using few demonstrations, while baselines require significantly more data to reach similar performance, particularly on more challenging tasks. The figure also includes an ablation where F_{bg} is trained using median-image-based supervision. While the average performance remains broadly comparable, the median-image-based variant occasionally exhibits less stable behavior and larger error cases.

Finally, we evaluated robustness to incomplete masks and imperfect temporal alignment on the Cube Stacking task across

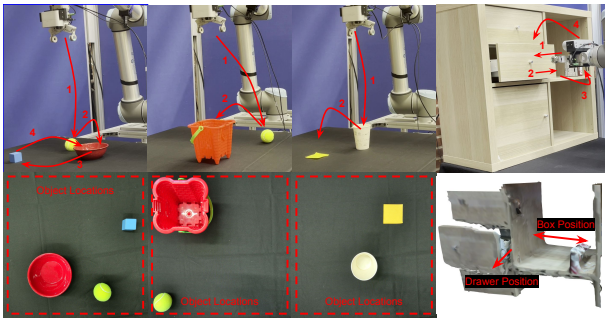

Fig. 7: Real-world task and scene variations. The top row illustrates the task steps, and the bottom row provides sample inputs, with rectangles and arrows marking possible scene parametrization. As occupancy values are hard to visually discern, we show the underlying mesh for the final task.

three seeds. The method maintained MED comparable to the original setup when up to 3 out of 30 demonstrations contained incomplete masks, while larger numbers prevented reliable object decomposition during scene-model training. For temporal alignment, we introduced temporal warping within sub-task segments while preserving key task phases. This increased the mean MED to 2.03 and reduced the task success rate to approximately 0.90, suggesting that the method retains reasonable performance under moderate intra-phase timing variation while still relying on coarse phase alignment across demonstrations.

B. Real-Robot Experiments

Validation of the framework is performed on four real-world robot tasks. These experiments demonstrate the method's robustness to perception noise, its ability to generalize across object categories using segmentation masks obtained from vision-language models, and its compatibility with 3D scene representations. These tasks are shown in Figure 7. The demonstrations are six-degree-of-freedom trajectories collected via kinesthetic teaching, except for the fourth task, where motion planning is used due to the complexity of the required movements. All demonstration trajectories are temporally aligned using gripper-state transitions (open/close), which serve as consistent semantic markers across demonstrations. During testing, we simplify motion execution by segmenting the predicted trajectories at gripper-state transitions (detected via rising and falling edges). Each sub-trajectory is then executed sequentially, and the corresponding gripper command is issued at the transition point. Trajectory execution is performed using MoveIt [31]. A small validation set is used for early stopping during the training of the motion generation model.

In the first task, the robot picks up tabletop objects and places them in a bowl, with varying locations across trials. It requires approximately 1–2 cm horizontal grasping accuracy to reliably grasp the tabletop objects. With 30 training demonstrations, the system successfully completes all 25 test cases [5]. We further evaluate robustness by adding visually distinct distractor objects in 5 additional test cases, all completed successfully. Representative input images and their corresponding reconstructions for this task are shown in Figure 8.

[5]With 20 training demonstrations, it completes 3 out of 5 additional trials. Failures occur when grasping the blue cube, which requires higher precision.

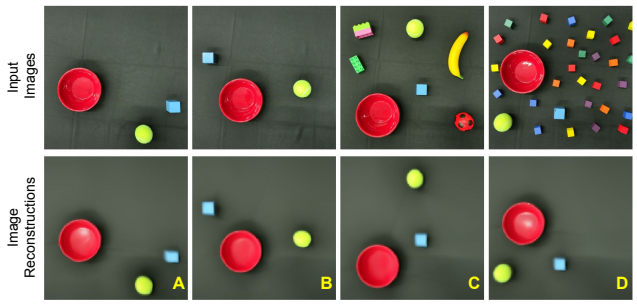

Fig. 8: Input Images, and corresponding image reconstructions with (C, D) and without (A, B) distractor objects for the first task.

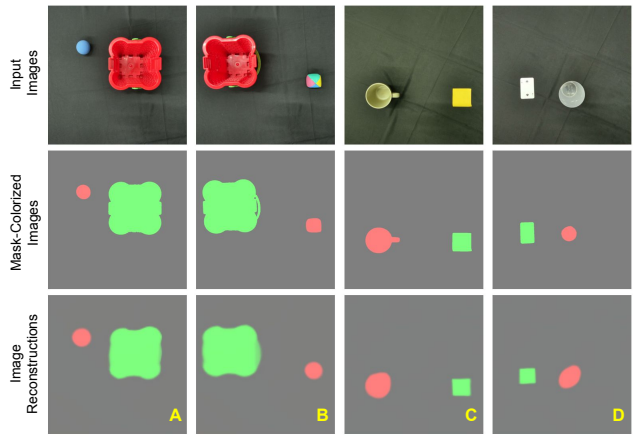

Fig. 9: Illustration of the perception pipeline: input images, language-conditioned mask colorizations produced by the vision–language model, and the resulting image reconstructions.

The second and third tasks evaluate category-level generalization using language-guided segmentation models. In the second task, the robot picks up a ball and places it in a container; in the third, it picks up a cup and places it at a target location marked by a piece of paper. Reliable grasping requires approximately 2 cm and 1 cm horizontal accuracy, respectively. We use a language-based segmentation algorithm [32] to detect object masks. The prompts "ball" and "box" (Task 2), and "cup" and "rectangle paper" (Task 3) are used to obtain segmentation masks, which are then used to generate colorized images and object masks. During training, the same objects are used across demonstrations. The models for the second and third tasks are trained with 20 and 30 demonstrations, respectively. At test time, we additionally evaluate on unseen objects from the same semantic categories. For the third task, these include two mugs and a watering can with similar cylindrical geometry and a top opening, allowing the demonstrated grasping strategy to transfer without modification. Figure 9 illustrates the perception pipeline, showing input images, language-conditioned colorized masks, and the corresponding reconstructions.

We evaluate both tasks on 15 test cases each. In the second task, the system succeeds in 14 out of 15 trials, while in the third task, it completes all test cases. The single failure in the second task occurs when testing with a previously unseen baseball, for which the model's predicted grasp was slightly misaligned; the harder surface of the baseball reduces compliance and imposes stricter precision requirements on the grasp. Under identical test conditions, but using the training

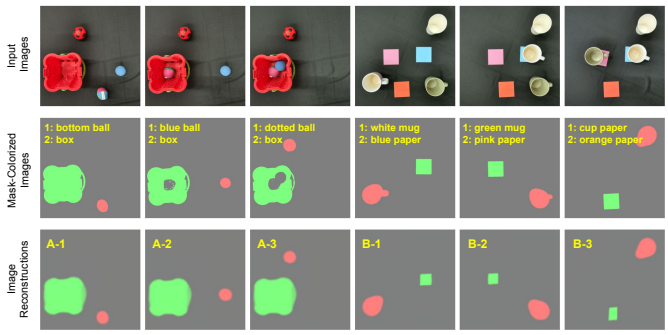

Fig. 10: Example multi-step trials for the second (A-1, A-2, A-3) and third (B-1, B-2, B-3) tasks. Language prompts in the second row allow the user to specify the object to be manipulated.

object (a tennis ball) instead of the baseball, the task succeeds, as the tennis ball's softer, more compliant surface reduces the precision required for successful grasping. We additionally evaluate a multi-object, language-conditioned setting for both tasks. For each trial, a language prompt (e.g., "tennis ball," "blue ball" for Task 2; "red paper," "playing card," "glass cup," "mug" for Task 3) specifies the target object. These prompts are used to obtain segmentation masks and corresponding colorized images [6]. The model then generates a trajectory for the specified object. We test this over five trials per task, and in each scenario, we perform three executions—each using different prompts and colorized images. Figure 10 shows representative scenes, prompts, colorized images, and reconstructions used for trajectory generation. Across all trials, the system successfully completes all three pick-and-place executions.

In the final task, the robot places a carton box inside a drawer. This task requires opening the drawer, retrieving the box from a neighboring compartment, and placing it inside the drawer, while the initial drawer opening and the box's initial position vary across trials. Grasping the drawer handle requires about 1 cm accuracy. This task tests the method's ability to generalize across full three-dimensional geometry and variations in scene surfaces. To represent the scene, we scan it from several predefined camera poses and estimate a Euclidean Signed Distance Field (ESDF) [33], which is then converted into an occupancy field by marking locations with ESDF $< \epsilon$ (with $\epsilon = 0.02$) as occupied. The model is trained using four such occupancy fields paired with their corresponding motion trajectories [7]. Evaluation on ten novel test configurations results in successful task completion in all cases. This level of generalization from only a few demonstrations is enabled by the object-centric representation, which captures the drawer opening and box position through interpolation between two latent corner vectors for each factor (open and closed for the drawer, and leftmost and rightmost positions for the box). Figure 11 shows the meshes acquired from scene scans together with the corresponding reconstructions obtained from the predicted occupancy fields, along with snapshots from task execution. The reconstructed

[6]This evaluation assumes that the language prompt produces a correct segmentation mask for the intended target object.

[7]With fewer than four examples, the scene representation model is unable to capture the drawer's articulation.

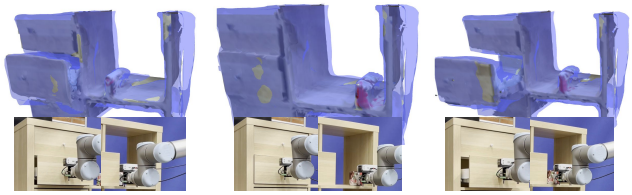

Fig. 11: Input meshes and reconstructions produced by the proposed method. Snapshots from robot motion corresponding to key grasp poses are shown right below the reconstructions. The reconstructed surfaces (shown in blue) capture the drawer openings and box positions, supporting motions that enable successful grasps of the drawer handle and the box.

meshes accurately reflect the positions of both the drawer and the box. The snapshots highlight the key poses that result in successful task execution. We further evaluate the model on eight held-out demonstrations collected under the same conditions as the training data. The average position error at key drawer-handle and box-grasp poses is 0.32 cm and 0.35 cm, respectively. The measured prediction errors indicate that the learned model maintains effective generalization across previously unseen configurations.

VI. Conclusion

This work presents a compositional framework for motion generation directly from visual demonstrations by coupling object-centric neural representations with a temporal mixture-of-experts model. The spatial mixture-of-experts learns smooth and compact latent representations that capture object-level variability from a small number of images, while the temporal mixture-of-experts exploits these latent representations to model motion as a sequence of object-conditioned primitives. The results demonstrate that compositional modeling at the object level provides a powerful paradigm for learning from demonstration, offering interpretable latent structure, data-efficient motion generation, and systematic generalization in both simulation and real-world environments. To support broader and more flexible generalization at the scene level, the framework will be extended to incorporate neural radiance field–based scene representations.

References

[1] S. Schaal, "Learning from demonstration," *Adv. Neural Inf. Process. Syst.*, vol. 9, 1996.

[2] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek, "Sequential composition of dynamically dexterous robot behaviors," *Int. J. Robot. Res.*, vol. 18, no. 6, pp. 534–555, 1999.

[3] Y. Du and L. P. Kaelbling, "Position: Compositional generative modeling: A single model is not all you need," in *Int. Conf. Mach. Learn. (ICML)*, 2024.

[4] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.

[5] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann, "Probabilistic movement primitives," *Adv. Neural Inf. Process. Syst.*, vol. 26, 2013.

[6] M. Y. Seker, M. Imre, J. H. Piater, and E. Ugur, "Conditional neural movement primitives." in *Robot.: Sci. Syst. (RSS)*, vol. 10, 2019.

[7] Y. Zhou, J. Gao, and T. Asfour, "Learning via-point movement primitives with inter-and extrapolation capabilities," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2019, pp. 4301–4308.

[8] G. Li, Z. Jin, M. Volpp, F. Otto, R. Lioutikov, and G. Neumann, "Prodmp: A unified perspective on dynamic and probabilistic movement primitives," *IEEE Robot. Autom. Lett.*, vol. 8, no. 4, pp. 2325–2332, 2023.

[9] A. Tekden, M. P. Deisenroth, and Y. Bekiroglu, "Neural field movement primitives for joint modelling of scenes and motions," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2023, pp. 3648–3655.

[10] Y. Yildirim and E. Ugur, "Conditional neural expert processes for learning movement primitives from demonstration," *IEEE Robot. Autom. Lett.*, 2024.

[11] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," *Intelligent service robotics*, vol. 9, pp. 1–29, 2016.

[12] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, vol. 44, no. 10-11, pp. 1684–1704, 2025.

[13] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *arXiv preprint arXiv:2304.13705*, 2023.

[14] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *Conference on Robot Learning*. PMLR, 2023, pp. 2165–2183.

[15] Y. J. Ma, J. Hejna, C. Fu, D. Shah, J. Liang, Z. Xu, S. Kirmani, P. Xu, D. Driess, T. Xiao *et al.*, "Vision language models are in-context value learners," in *International Conference on Learning Representations*, vol. 2025, 2025, pp. 33 984–34 009.

[16] P. Li, Y. Wu, Z. Xi, W. Li, Y. Huang, Z. Zhang, Y. Chen, J. Wang, S.-C. Zhu, T. Liu *et al.*, "Controlvla: Few-shot object-centric adaptation for pre-trained vision-language-action models," in *Conference on Robot Learning*. PMLR, 2025, pp. 1898–1913.

[17] F. Locatello *et al.*, "Object-centric learning with slot attention," *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, pp. 11 525–11 538, 2020.

[18] M. Niemeyer and A. Geiger, "Giraffe: Representing scenes as compositional generative neural feature fields," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 11 453–11 464.

[19] Y. Xie *et al.*, "Neural fields in visual computing and beyond," *Comput. Graph. Forum*, 2022.

[20] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *European conference on computer vision*. Springer, 2020, pp. 405–421.

[21] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 165–174.

[22] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4460–4470.

[23] A. Simeonov *et al.*, "Neural descriptor fields: Se (3)-equivariant object representations for manipulation," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2022, pp. 6394–6400.

[24] A. Tekden, M. P. Deisenroth, and Y. Bekiroglu, "Grasp transfer based on self-aligning implicit representations of local surfaces," *IEEE Robot. Autom. Lett.*, vol. 8, no. 10, pp. 6315–6322, 2023.

[25] A. Tekden, D. Kanoulas, A. Billard, and Y. Bekiroglu, "Reactive motion generation via phase-varying neural potential functions," *IEEE Robotics and Automation Letters*, 2026.

[26] H.-T. D. Liu, F. Williams, A. Jacobson, S. Fidler, and O. Litany, "Learning smooth neural functions via lipschitz regularization," in *ACM SIGGRAPH Conf.*, 2022, pp. 1–13.

[27] A. Ude, B. Nemec, T. Petrić, and J. Morimoto, "Orientation in cartesian space dynamic movement primitives," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2014, pp. 2997–3004.

[28] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *AAAI Conf. Artif. Intell. (AAAI)*, vol. 32, no. 1, 2018.

[29] D. Ha, A. M. Dai, and Q. V. Le, "Hypernetworks," in *Int. Conf. Learn. Represent. (ICLR)*, 2017.

[30] O. Siméoni *et al.*, "DINOv3," 2025.

[31] D. T. Coleman, I. A. Sucan, S. Chitta, and N. Correll, "Reducing the barrier to entry of complex robotic software: a moveit! case study," *J. Softw. Eng. Robot.*, vol. 5, no. 1, pp. 3–16, 2014.

[32] L. Medeiros, "Lang segment anything," GitHub repository, 2023, https://github.com/luca-medeiros/lang-segment-anything.

[33] A. Millane *et al.*, "nvblox: Gpu-accelerated incremental signed distance field mapping," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2024, pp. 2698–2705.