

# SEM: Enhancing Spatial Understanding for Robust Robot Manipulation

Xuewu Lin, Tianwei Lin, Yun Du, Yiwei Jin, Jitao Li, Hongyu Xie, Lichao Huang, Zhizhong Su\*

**Abstract**—A key challenge in robot manipulation lies in developing policy models with consistent spatial understanding—the ability to reason about 3D geometry, object relations, and robot state. Existing mainstream models take 2D images as input, without performing explicit 3D modeling, and thus lack spatial understanding capabilities as well as 3D and embodiment generalization. To address this, we propose SEM (Spatial Enhanced Manipulation), a diffusion-based policy framework that constructs a unified spatial representation by projecting multi-view image features and joint-centric robot states into a unified 3D space. This spatially aligned representation provides a consistent feature space for the diffusion policy to condition on during action generation. Extensive experiments demonstrate that SEM significantly improves spatial understanding, leading to robust and generalizable manipulation across diverse tasks that outperform existing baselines.

## I. INTRODUCTION

Robust robot manipulation in diverse real-world environments hinges on a core capability: spatial understanding—the ability to perceive 3D geometry, object relations, and the robot’s own embodiment states in a consistent way. While recent diffusion-based (or flow-based) policy models have demonstrated strong generative capacity for action modeling, their effectiveness remains limited by perceptual bottlenecks: (1) vision backbones are typically trained for 2D image understanding (visual grounding and visual question answering) and lack explicit 3D reasoning ability, (2) visual features and robot state features are processed independently without spatial alignment, and (3) the robot’s embodiment structure is not effectively modeled.

To address these challenges, we propose SEM (Spatial Enhanced Manipulation), a policy framework that constructs a unified spatial representation as a consistent representation space for action generation (Fig. 1). First, we introduce a 3D spatial enhancer that injects explicit 3D information into 2D visual features. This module encodes camera intrinsic and extrinsic parameters into the visual representation via 3D absolute position embeddings. Second, when encoding the robot state, we augment the joint angles with the 6D poses, obtained via forward kinematics, and embed these poses as part of the robot state features. Both the 3D absolute position of visual features and the joint poses are expressed in a unified 3D coordinate frame, such as the robot base frame. Finally, to better capture the embodiment structure and support robots with arbitrary numbers of joints, we design a joint-centric robot state encoder and action decoder, which enable feature propagation among joints through graph attention.

\*Horizon, Beijing, China, zhizhong.su@horizon.auto

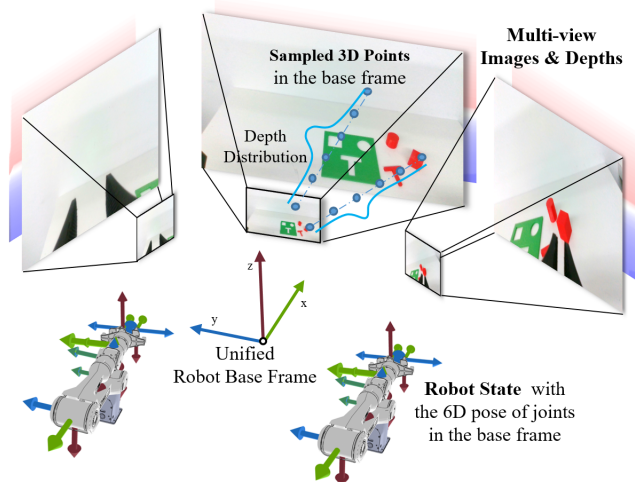


Fig. 1. Overview of SEM. SEM constructs a unified spatial representation by projecting multi-view image features and joint-level robot states into the robot base frame.

Robot embodiments and sensor configurations often differ significantly across datasets. When training policies on such heterogeneous data, severe ambiguities can arise: the same joint index may correspond to distinct physical motions on different robots, and varying camera setups can yield observations from inconsistent viewpoints. These ambiguities are a major obstacle to embodiment and sensor generalization in existing models. These ambiguities are the key factor limiting the embodiment and sensor generalization capabilities of existing models. SEM mitigates such inconsistencies by explicitly modeling both the camera and the robot embodiment, thereby achieving 3D and embodiment generalization and obtaining improved performance.

We extensively evaluate SEM in both simulation and real-robot settings. In simulation, we adopt RoboTwin 2.0 benchmark [1] and evaluate on 16 representative tasks. Despite having only around 40M trainable parameters, SEM significantly outperforms existing approaches. We further conduct targeted experiments on 3D spatial generalization and embodiment generalization, confirming the benefits of our unified spatial representation. In real-world evaluations, SEM consistently outperforms RDT [2] across all seven tasks. Our contributions are summarized as follows:

- **Novel Framework.** We propose SEM, a new VLA framework, from the perspective of enhancing spatial understanding.
- **Superior Performance.** We evaluate SEM thoroughly in both simulation and real-world robot settings, demonstrat-

ing that SEM outperforms existing baselines.

- **Comprehensive Analysis.** We verify the effectiveness of each module in SEM, as well as its improvement in 3D spatial and embodiment generalization.

## II. RELATED WORKS

End-to-end robot manipulation methods have progressed rapidly, with improvements being made from various perspectives. Early works [3], [4], [5], [6] use imitation learning to acquire manipulation skills from human demonstrations. Subsequent studies refine action modeling to improve stability and performance: IBC [7] introduces implicit behavioral cloning, BeT [8] models multi-modal actions with a Gaussian mixture, and Diffusion Policy (DP) [9] formulates trajectory generation as a conditional diffusion process, enabling robust learning in high-dimensional action spaces.

Moreover, many works have contributed to the improvement of algorithmic pipelines and network architectures. PerAct [10] encodes voxelized 3D features with a Perceiver transformer to predict end-effector (EE) keyframes. Act3D [11] introduces coarse-to-fine 3D point sampling for precise action prediction. 3D Diffuser Actor [12] combines Act3D with diffusion modeling for further gains. RVT [13] re-renders observations into three orthogonal planes to produce heatmaps of EE keyframes, while RVT-2 [14] adds a fine branch to boost accuracy. KStarDiffuser [15] uses graph convolution over joint trajectories with an auxiliary joint position loss for bimanual tasks. However, all these approaches predict EE poses, which limits their ability to handle complex long-horizon or contact-rich tasks such as cloth manipulation. DP3 [16] addresses this by directly predicting joint positions via diffusion, enabling finer control.

Recently, breakthroughs in VLMs [17], [18], [19], [20] have spurred growing research interest in generalist action models. These models [21], [22], [2], [23], [24], [25], [26] use VLMs as the base model and are trained on diverse embodied intelligence datasets with the aim to achieve high levels of scenario, embodiment, and task generalization. Methods such as SpatialVLA [27] enhance VLMs with 3D position embeddings for improved spatial reasoning. Despite these advances, VLM-based policies still face challenges in real-world performance and spatial generalization.

Our work focuses more on network design for improved 3D spatial understanding rather than concentrating on VLM-based generalist models. The most closely related approaches are KStarDiffuser [15] and SpatialVLA [27]. Unlike SEM, both of them output EE poses rather than joint trajectories, and SpatialVLA does not model camera extrinsic or unify visual and action representations in a common 3D space.

## III. PROBLEM FORMULATION

The inputs for the manipulation problem primarily consist of sensor data and current joint positions  $S$ , where the sensor data generally encompass multi-view images  $I$  and depth maps  $D$ . To simplify the problem, this paper utilizes only the sensor data from the current frame. When handling multitasks, instruction texts  $T$  are also required. Moreover,

we consider modeling both the camera and the embodiment structure, thus requiring inputs of camera parameters  $C$  and embodiment structure parameters  $E$ . Thus, the inputs can be summarized into the following parts:

$$\{I \in \mathbb{R}^{N \times H \times W \times 3}, D \in \mathbb{R}^{N \times H \times W \times 1}, S \in \mathbb{R}^{N_j}, T\} \cup \{C, E\} \quad (1)$$

Here,  $N$  denotes the number of images,  $N_j$  represents the number of joints. For a standard pinhole camera,  $C$  represents the camera intrinsic and extrinsic matrix. The output is the robot joint positions for the subsequence  $t_{out}$  frames, denoted as  $S_{out} \in \mathbb{R}^{N_j \times t_{out}}$ .

## IV. OUR METHOD: SEM

The overall architecture of SEM, illustrated in Fig.2, comprises five sub-modules: a vision language model (VLM) serving as a backbone for extracting features from images and text; a depth encoder for encoding 2D depth maps; a spatial enhancer that injects spatial and geometric awareness into 2D image tokens (Section IV-B); a robot state encoder used for encoding the robot’s historical trajectory  $S$  (Section IV-C); and an action decoder that predicts future trajectories  $S_{out}$  from the aggregated features (Section IV-D).

### A. Backbone: VLM

SEM can be integrated with any existing VLM to perform image and text feature encoding and leverage its strong semantic understanding capability. In this paper, we use two different types of VLMs: the 2D detection foundation model GroundingDINO and the LLM-based Qwen-2.5-VL. Note that we consider GroundingDINO as a VLM in a broad sense. The image features output by the VLM are denoted as  $F^I$ , and the text features are denoted as  $F^T$ .

### B. Spatial Enhancer

To effectively leverage both rich 2D image features and accurate 3D geometric information, we design a spatial enhancer within SEM. In multi-view feature fusion tasks such as 3D detection, a common practice (PETR [28], BIP3D [29]) is to aggregate features based on sampled points along the camera frustum. Inspired by this, our spatial enhancer integrates multi-view 2D image features with depth-aware 3D representations to enhance 3D spatial understanding.

In this work, the proposed spatial enhancer performs two critical functions: (1) establishing a camera model to project 2D image features into 3D space, and (2) fusing image features with depth features to produce representations with stronger geometric awareness. As illustrated in Fig. 3(a), the enhancer takes multi-view image features, depth features, and camera parameters as input, and outputs enhanced features with better spatial understanding. Specifically, for each image feature  $F_{i,j}^I$ ,  $K$  depth values is sampled, and the corresponding pixel coordinates are projected into 3D space using the camera model.

$$P_{i,j} = \{T_e^{-1}T_i^{-1}[i \times d_k, j \times d_k, d_k]^T \in \mathbb{R}^3 | 1 \leq k \leq K\} \quad (2)$$

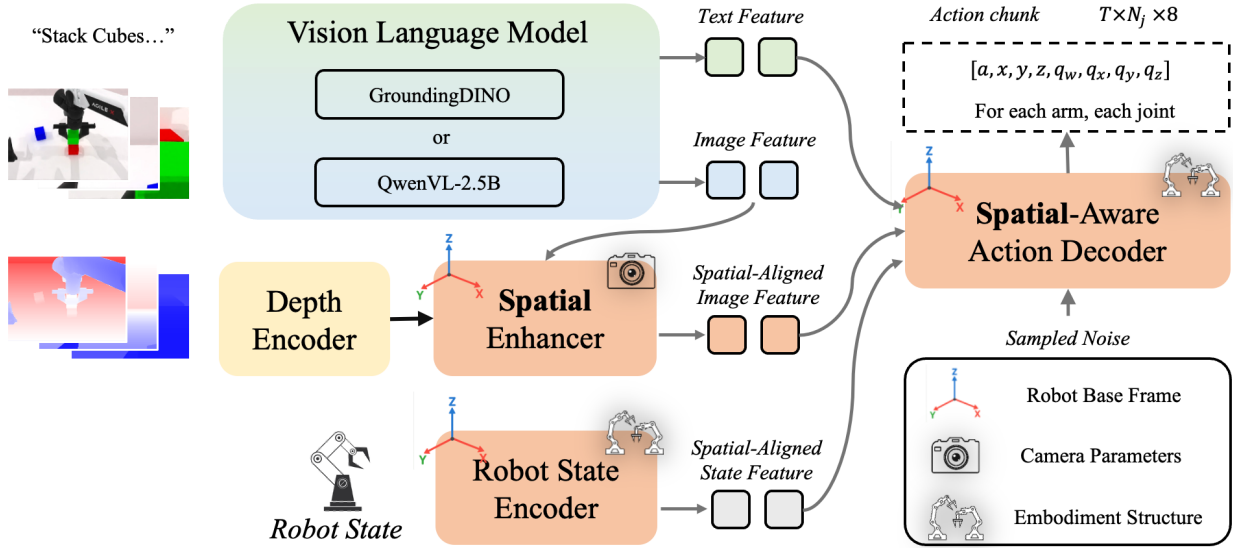


Fig. 2. The overall architecture of SEM. SEM takes multi-view images, depth, robot state, and instructions as inputs, projects them into a shared base-frame 3D space, and generates future joint trajectories via a diffusion-based policy. In this way, SEM learns a spatially consistent representation that improves performance and generalization.

Here  $T_e$  and  $T_i$  are the extrinsic and intrinsic matrix,  $i$  and  $j$  are the pixel coordinates of image features. These 3D points  $P$  are then encoded into high-dimension.

$$E_{i,j,k}^P = \text{MLP}(P_{i,j,k}) \quad (3)$$

Next, a discrete depth distribution  $W$  is predicted and used to compute a weighted sum over these 3D points embeddings, resulting in a 3D positional embedding  $E_{i,j}$  for each image feature.

$$W_{i,j} = \text{MLP}(F_{i,j}^I, F_{i,j}^D) \in \mathbb{R}^K \quad (4)$$

$$E_{i,j} = W_{i,j} \times E_{i,j}^P \quad (5)$$

$F^D$  denotes the depth features output by the depth encoder. Finally, the image feature, depth feature, and 3D positional embedding are fused via a simple MLP.

$$F_{i,j}^{3D} = \text{MLP}(F_{i,j}^I, F_{i,j}^D, E_{i,j}) \quad (6)$$

To support scenarios without depth sensors, the depth feature is designed to be pluggable. When depth input is unavailable, we directly predict the depth distribution from image features, similar to monocular depth estimation. For 3D encoding, we adopt the robot's base coordinate frame to unify multi-view features. Subsequent 3D encodings for the embodiment are performed consistently in this coordinate frame, as detailed in Sections IV-C and IV-D.

### C. Robot State Encoder

The robot state encoder is used to encode the current robot state  $S$ . A state representation containing only joint angles is insufficient, as the same joint angles may correspond to different physical states across embodiments. To capture more informative information, we compute the 6D pose of each joint in the robot's base coordinate frame via

forward kinematics and concatenate these poses with the joint positions to form the enhanced state representation  $S'$ .

$$S' \in \mathbb{R}^{N_j \times 8}, S'_i = [a, x, y, z, q_w, q_x, q_y, q_z]_i \quad (7)$$

where  $a$  is the joint angle,  $[x, y, z]$  represents the 3D coordinates of the joint, and  $[q_w, q_x, q_y, q_z]$  denotes the rotation quaternion. We project the feature  $S'$  into a high-dimensional space using an MLP, then feed it into a transformer encoder with joint graph attention as its core component. The joint graph attention utilizes the joint distance matrix  $J \in \mathbb{R}^{N_j \times N_j \times 1}$  to encode relative positions, where each entry  $J_{i,j}$  denotes the number of links between the  $i_{th}$  query joint and the  $j_{th}$  key joint. The formulation is as follows.

$$P = \text{MLP}(\text{SinCos}(J)) \in \mathbb{R}^{N_j \times N_j \times d} \quad (8)$$

$$A = \text{einsum}(ik, ijk, jk \rightarrow ij | Q, P, K) \in \mathbb{R}^{N_j \times N_j} \quad (9)$$

$$O = \text{softmax}\left(\frac{A}{\sqrt{d}}\right) V \quad (10)$$

where  $Q$ ,  $K$ ,  $V$  and  $O$  denote query, key, value and output features,  $d$  is the feature channels, and  $\text{einsum}$  is the Einstein summation convention. It is important to note that the computational complexity of joint graph attention is  $O(N_j^2 d^2)$ , which is higher than vanilla attention's  $O(N_j^2 d + N_j d^2)$ . However, since  $N_j$  is typically small, the computational load of this module is not a bottleneck.

It can be observed that our robot state encoder is joint-centric and naturally generalizes to embodiments with an arbitrary number of joints, which is crucial for training with mixed multi-embodiment data.

### D. Spatial-Aware Action Decoder

The action decoder takes all condition features (image features, text features, and robot state features) as input and employs a diffusion transformer to predict future trajectories.

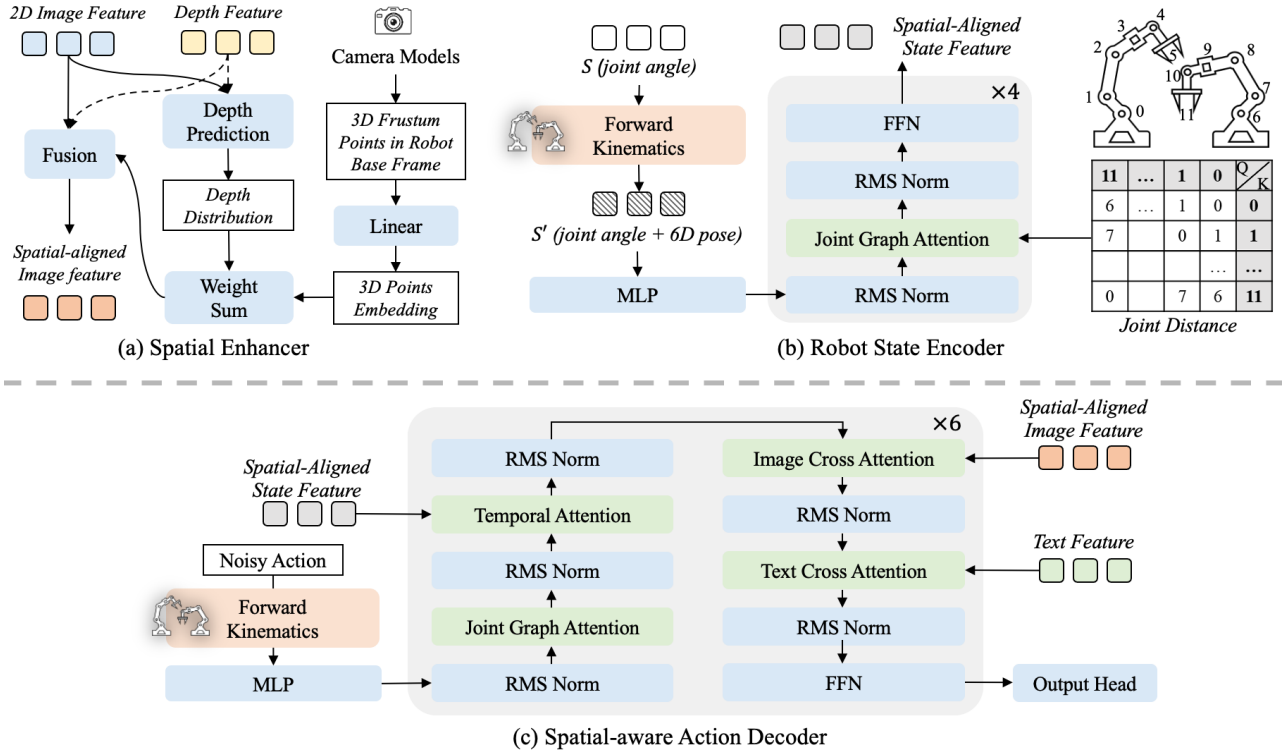


Fig. 3. Details of SEM modules. (a) Spatial enhancer lifts multi-view features into the base-frame 3D space. (b) Robot state encoder embeds joint angles and 6D poses with graph attention to model embodiment structure. (c) Action decoder predicts future joint trajectories via diffusion.

First, we sample noise on the joint angles. Then, similar to the robot state encoder, we use forward kinematics to obtain the noisy robot state, denoted as  $S'_{noise} \in \mathbb{R}^{N_j \times t_{out} \times 8}$ . In terms of network architecture, the action decoder shares a joint-centric design with the state encoder, utilizing joint graph attention to enable feature interaction among joints. We employ three independent cross-attention layers that attend to robot state, image, and text features respectively (see Fig. 3(c)). The temporal cross-attention operates in a causal manner, ensuring a unidirectional flow along the time axis. Finally, an output head consisting of upsampling and 1D convolution layers generates the structured prediction  $S'_{pred} \in \mathbb{R}^{N_j \times t_{out} \times 8}$ , which can be further refined via denoising steps. For motion control, we use only the predicted joint angles in  $S'_{pred}$ , while the predicted 6D poses just serve as auxiliary supervision. Nevertheless, future work could explore leveraging the predicted 6D poses with inverse kinematics for control.

### E. Training Loss

During training, we optimize four loss terms: joint position loss, joint pose loss, forward kinematics pose loss, and depth loss, as defined in Equation 11.

$$L = \lambda_1 L_{joint} + \lambda_2 L_{pose} + \lambda_3 L_{pose}^{fk} + \lambda_4 L_{depth} \quad (11)$$

Here, the joint position loss  $L_{joint}$  and joint pose loss  $L_{pose}$  are defined as the L1 distances between the predicted joint angles and 6D poses and their respective ground truths. To further improve positional accuracy, the forward kinematics

pose loss  $L_{pose}^{fk}$  recomputes 6D poses from the predicted joint angles using forward kinematics and compares them with the ground truth via an additional L1 distance. Unlike prior methods, our approach simultaneously constrains joint angles and 6D poses, providing stronger and more informative supervision, which leads to improved performance. The depth loss  $L_{depth}$ , applied to the depth distribution predicted by the spatial enhancer, is formulated as a cross-entropy loss.

## V. EXPERIMENTS

### A. Benchmark

We evaluate SEM in both simulation and real-world settings. In simulation, we conduct experiments on the RoboTwin 2.0 platform [1] using the aloha-agilex embodiment. From RoboTwin’s task suite, we select 16 representative tasks based on diversity, difficulty, and expert trajectory synthesis success rate. The selected tasks are listed in the Tab.II. To assess SEM’s generalization capabilities, we design two additional evaluations: (1) a 3D spatial generalization test and (2) an embodiment generalization test. Both are conducted in simulation, as illustrated in Fig. 4(b,c), with details provided in Sec. V-D and Sec. V-E. Real-world experiments are presented separately in Sec. V-G.

### B. Implementation Details

We adopt Grounding DINO Tiny [30] and Qwen2.5-VL-3B [31] as the VLM backbone. When using Qwen2.5-VL-3B, we freeze its parameters and train only the newly added modules. The parameter distributions of both configurations are shown in Tab.I. Compared with most existing VLA

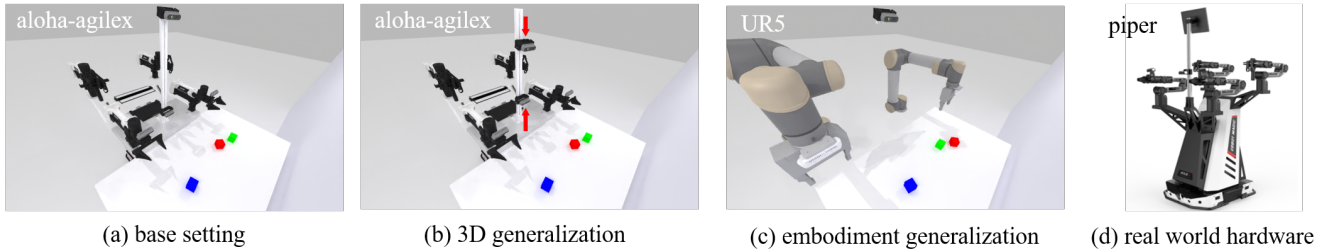


Fig. 4. We use RoboTwin2.0 as the simulation benchmark. The base environment features the AgileX-Aloha embodiment, equipped with two environment cameras and two hand cameras, on a clean table. For the 3D generalization experiments, we adjust the height of the two environment cameras. For the embodiment generalization experiments, we replace the test robot with two UR5 arms. In real-world testing, we use the Piper.

TABLE I

MODEL PARAMETER COUNTS (M). GD DENOTES GROUNDINGDINO TINY, QW REPRESENTS QWEN2.5-VL-3B. PARAMETERS MARKED WITH UNDERSCORES REMAIN FROZEN DURING TRAINING.  $\mathcal{E}$ ,  $E$ , AND  $D$  DENOTE THE ENCODERS, ENHANCERS, AND DECODER RESPECTIVELY.

Model	VLM	$\mathcal{E}_{depth}$	$\mathcal{E}_{state}$	$E_{spatial}$	$D_{action}$	Trainable	All
SEM - GD	153.00	0.09	5.77	0.73	15.55	175.33	175.33
SEM - QW	<u>3580.69</u>	0.74	5.77	1.25	15.55	41.33	3622.02

models [32], [22], [2], SEM uses a lightweight action decoder with substantially fewer parameters. This aligns with a hierarchical design philosophy: the VLM backbone acts as the "brain," providing semantic understanding and reasoning, while the action decoder is responsible for instruction following, and thus requires comparatively fewer parameters.

For training, we use 160k steps for multi-task models and 80k steps for single-task models, with 256 batch size. For the diffusion process, we adopt DDPM [33] with 1000 timesteps and use DPM-Solver [34] for fast inference with 10 denoising steps. We directly predict the noisy sample rather than noise, which empirically accelerates convergence.

### C. Main Results

We first evaluate our method under the single-task setting, with each model trained on 50 demonstrations per task. As shown in Tab. II, SEM achieves significant performance gains, improving average success rates by 22.13% over DP3 [16], 35.38% over ACT [35] and 47.32% over DP [9]. Compared to two well-pretrained models, RDT [2] and Pi0 [22], SEM also shows significant advantages in the simulation environment. Notably, SEM demonstrates especially strong performance on tasks requiring high-precision manipulation, achieving an 80.0% success rate on the 'block ranking rgb' task, compared to 3.0% (DP3) and 0.0% (DP).

We further assess performance in multi-task scenarios, training each model on 1,600 demonstrations (100 per task). We evaluated two variants of our model, SEM-GD and SEM-QW, achieving success rates of 84.5% and 84.62%, respectively, both significantly outperforming the baseline RDT-1B (55.12%). Meanwhile, our trainable parameters are also considerably fewer than RDT-1B.

### D. 3D Generalization Test

To demonstrate that SEM possesses strong 3D generalization capabilities, we design the following experiment. We adjusted the height of the two environment cameras, as

shown in Fig.4(b), and then conduct zero-shot testing of the multi-task model SEM-QW and RDT under this modified camera setup. SEM consistently gets higher average success rates across 16 tasks compared to RDT, and its normalized success rate exhibits a slower rate of decay as camera height varies, as shown in Fig.5(left).

Since some tasks (e.g., handover mic and open laptop) are not sensitive to camera height variations, and the presence of hand cameras may influence the experimental conclusions, we also train two single-task (place empty cup) models with a single environment camera, and test them under different camera heights. SEM also demonstrate superior 3D generalization capabilities in this setting, as shown in Fig.5(right).

### E. Embodiment Generalization Test

To evaluate the embodiment generalization capability, we train on a mixed dataset consisting of data from two robot embodiments: AgileX-Aloha and Dual-UR5. Specifically, it includes 100 demonstrations per task for AgileX-Aloha and 50 demonstrations per task for UR5. As a comparison, we train the model using only UR5 data, with 50 demonstrations per task. If the model possesses embodiment generalization ability, the inclusion of AgileX-Aloha data is expected to improve its performance on the UR5. The results, shown in Tab.IV, indicate that training with the mixed dataset leads to a 5.5% performance improvement for SEM compared to training with UR5 data only, whereas RDT exhibits a performance drop of 26.69%. SEM possesses strong embodiment generalization capability, whereas models like RDT can be negatively affected by the ambiguity of multi-embodiment data. In addition, we finetune AgileX-pretrained SEM on the UR5 dataset and observe a significant improvement in convergence speed, as shown in Fig.6.

TABLE II

ROBOTWIN2.0 BENCHMARK RESULTS. \* INDICATES MULTI-TASK MODELS, WHILE OTHERS ARE SINGLE-TASK MODELS. SINGLE-TASK RESULTS OF BASELINES ARE ALL FROM THE OFFICIAL ROBOTWIN LEADERBOARD.

	adjust bottle	beat block hammer	blocks ranking rgb	blocks ranking size	dump bin bigbin	handover mic
DP [9]	97.00	42.00	0.00	1.00	49.00	53.00
RDT [2]	81.00	77.00	3.00	0.00	64.00	90.00
ACT [35]	97.00	56.00	1.00	0.00	68.00	85.00
Pi0 [22]	90.00	43.00	19.00	7.00	83.00	98.00
DP3 [16]	99.00	72.00	3.00	2.00	85.00	100.00
SEM-QW	100.00	86.00	80.00	12.00	96.00	100.00
RDT*	98.00	84.00	20.00	3.00	92.00	100.00
SEM-GD*	98.00	100.00	66.00	40.00	100.00	100.00
SEM-QW*	100.00	96.00	82.00	34.00	98.00	100.00
	lift pot	move pillbottle pad	open laptop	open microwave	place cans plasticbox	place dual shoes
DP	39.00	1.00	49.00	5.00	40.00	8.00
RDT	72.00	8.00	59.00	37.00	6.00	4.00
ACT	88.00	0.00	56.00	86.00	16.00	9.00
Pi0	84.00	21.00	85.00	80.00	34.00	15.00
DP3	97.00	41.00	82.00	61.00	48.00	13.00
SEM-QW	92.00	62.00	84.00	96.00	84.00	48.00
RDT*	80.00	13.00	61.00	59.00	16.00	14.00
SEM-GD*	98.00	68.00	82.00	82.00	100.00	78.00
SEM-QW*	96.00	70.00	86.00	98.00	94.00	58.00
	place empty cup	rotate qrcode	stack blocks three	stack bowls three	Mean	Gain
DP	37.00	13.00	0.00	63.00	31.06	
RDT	56.00	50.00	2.00	51.00	41.25	
ACT	61.00	1.00	0.00	48.00	42.00	
Pi0	37.00	68.00	17.00	66.00	52.94	
DP3	65.00	74.00	1.00	57.00	56.25	
SEM-QW	96.00	84.00	54.00	80.00	78.38	<b>+22.13</b>
RDT*	79.00	82.00	17.00	64.00	55.12	
SEM-GD*	94.00	86.00	80.00	80.00	84.50	<b>+29.38</b>
SEM-QW*	98.00	88.00	72.00	84.00	84.62	<b>+29.50</b>

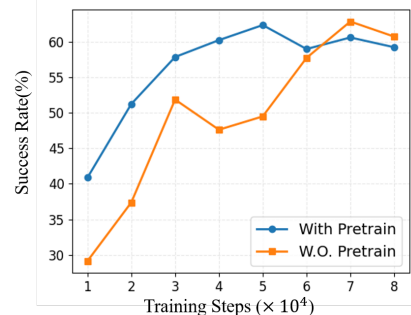
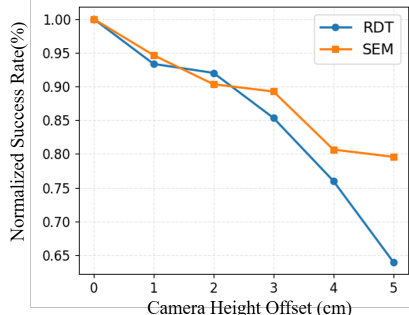
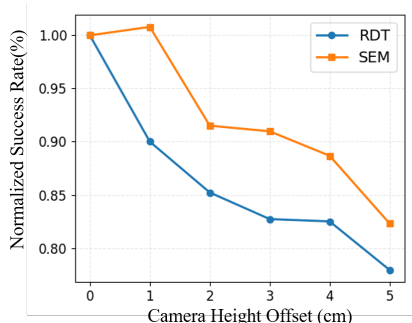


Fig. 5. 3D Generalization Test Results. The normalized success rate is defined as the ratio of at each camera height offset to at 0 cm. Left: Multi-task & multi-view. Right: Single-view, single-task (place empty cup).

Fig. 6. Embodiment Generalization Results. Performance on UR5, with and without pretraining on Agilix.

### F. Ablation Study

We first conduct an ablation study on the spatial enhancer. This module serves two main functions: fusing depth features and performing 3D encoding. Removing only the depth features reduced the average success rate by 8.5%, and further removing the entire spatial enhancer led to a total reduction of 15.72%, as shown in Tab.III (id 1 & 2).

We also ablate the unified 3D space. Replacing the image 3D position embedding from the unified base frame with the camera frame reduces performance by 3.37% (Tab.III, id 3).

Last, we evaluated the role of the joint graph attention by replacing it with vanilla attention, which resulted in a 4.62%

reduction (Tab.III, id 4). It is important to highlight that joint graph attention is expected to be even more pronounced when trained on more diverse embodiment data, which we aim to verify in future work.

### G. Real World Experiments

For the real-world experiments, we evaluate SEM on seven tasks. Five tasks are selected from RoboTwin: as shown in Fig. 7 (c-g). In addition, we introduce two novel tasks: *place blocks into slots*, designed to evaluate fine-grained manipulation capability, and *fold towel*, which evaluates performance on deformable object manipulation. The experimental setup

TABLE III

ABLATION STUDY OF SEM. 'SE' AND 'JGA' DENOTE 'SPATIAL ENHANCER' AND 'JOINT GRAPH ATTENTION', RESPECTIVELY.

id	setting	adjust bottle	beat block hammer	blocks ranking rgb	blocks ranking size	dump bin bigbin	handover mic
1	No Depth & SE	100.00	90.00	44.00	4.00	90.00	100.00
2	No Depth	100.00	96.00	76.00	24.00	94.00	100.00
3	No Unified 3D	100.00	88.00	84.00	28.00	92.00	100.00
4	No JGA	100.00	98.00	78.00	46.00	92.00	100.00
	Full Model	100.00	96.00	82.00	34.00	98.00	100.00
id	setting	lift pot	move pillb. pad	open laptop	open micro.	place cans plast.	place shoes
1	No Depth & SE	82.00	20.00	70.00	56.00	44.00	32.00
2	No Depth	84.00	48.00	80.00	86.00	80.00	44.00
3	No Unified 3D	96.00	50.00	88.00	96.00	100.00	58.00
4	No JGA	94.00	58.00	80.00	66.00	100.00	58.00
	Full Model	96.00	70.00	86.00	98.00	94.00	58.00
id	setting	place empty cup	rotate qrcode	stack blocks three	stack bowls three	Mean	Degradation
1	No Depth & SE	72.00	80.00	34.00	64.00	58.80	<b>-15.72</b>
2	No Depth	94.00	80.00	56.00	76.00	76.12	<b>-8.50</b>
3	No Unified 3D	94.00	92.00	50.00	84.00	81.25	<b>-3.37</b>
4	No JGA	98.00	86.00	50.00	76.00	80.00	<b>-4.62</b>
	Full Model	98.00	88.00	72.00	84.00	84.62	

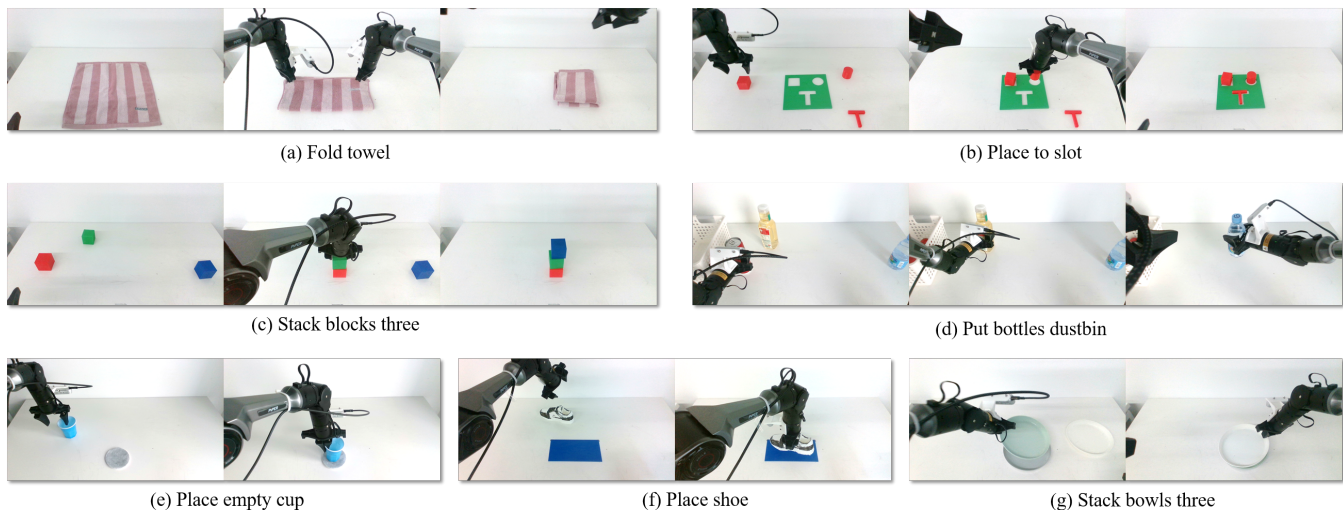


Fig. 7. Examples of real-world tasks. (a) deformable object manipulation task, (b) fine manipulation task, (c–g) same as RoboTwin settings.

TABLE IV

EMBODIMENT GENERALIZATION RESULTS. TRAINING WITH MULTI-EMBODIMENT DATA YIELDS PERFORMANCE GAINS FOR SEM, BUT HAS A SIGNIFICANT NEGATIVE EFFECT ON RDT.

	SEM	RDT
Only UR5	60.75	63.00
Agilex pretrain, UR5 finetune	59.25 ( <b>-1.50</b> )	46.30 ( <b>-16.70</b> )
Mix Agilex and UR5	66.25 ( <b>+5.50</b> )	36.31 ( <b>-26.69</b> )

is shown in Fig. 7. We conduct two sets of comparative experiments: (1) both SEM and RDT are first pretrained on RoboTwin simulation data and then post-trained on real-world demonstrations for a fair comparison; (2) following common practice, SEM is pretrained on the open-source real-world dataset (Agibot-world we used) and post-trained on our data, while RDT uses its open-source pretrained weights. As summarized in Tab.V, SEM significantly outperforms RDT whether pretrained on simulation or real-world data.

## VI. CONCLUSION AND FUTURE WORKS

This work introduces SEM, a vision-language-action framework that builds a unified spatial representation by projecting multi-view image features and robot states into a shared 3D base frame. By explicitly modeling camera parameters and embodiment structure, SEM enables robust, generalizable end-to-end motion planning. Extensive experiments in both simulation and real-world settings, along with ablation studies, validate the effectiveness of SEM in improving 3D spatial and embodiment generalization.

Despite these results, SEM still has room for further improvement. Future work includes refining the definition of relative joint distances, integrating more robust and accurate depth prediction model, pretraining on larger and more diverse datasets, leveraging VLMs for reasoning over long-horizon tasks, and exploring reinforcement learning to improve policy robustness and reliability.

TABLE V

TASK PROGRESS IN THE REAL-WORLD TESTING. THE “+” INDICATES MODELS PRETRAINED ON OPEN-SOURCE REAL-WORLD DATA, WHILE THE OTHERS WERE PRETRAINED ON ROBOTWIN SIMULATION DATA.

	Task a	Task b	Task c	Task d
RDT	-	-	44.0	36.0
SEM	-	-	66.0	38.0
RDT <sup>+</sup>	41.3	68.0	64.0	33.0
SEM <sup>+</sup>	68.4	69.0	78.0	68.0
	Task e	Task f	Task g	Mean
RDT	42.7	18.0	65.3	41.2
SEM	63.0	34.0	66.7	53.5 (+12.3)
RDT <sup>+</sup>	38.0	28.0	55.0	46.8
SEM <sup>+</sup>	64.7	64.7	95.3	72.6 (+25.8)

## REFERENCES

- [1] T. Chen, Z. Chen, B. Chen, Z. Cai, Y. Liu, Q. Liang, Z. Li, X. Lin, Y. Ge, Z. Gu *et al.*, “Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation,” *arXiv preprint arXiv:2506.18088*, 2025.
- [2] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu, “Rdt-1b: a diffusion foundation model for bimanual manipulation,” in *The Thirteenth International Conference on Learning Representations*.
- [3] D. A. Pomerleau, “Alvin: An autonomous land vehicle in a neural network,” *Advances in neural information processing systems*, vol. 1, 1988.
- [4] F. Torabi, G. Warnell, and P. Stone, “Behavioral cloning from observation,” in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2018.
- [5] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, “Deep imitation learning for complex manipulation tasks from virtual reality teleoperation,” in *2018 IEEE international conference on robotics and automation (ICRA)*. Ieee, 2018, pp. 5628–5635.
- [6] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. Funkhouser, “Tossing-bot: Learning to throw arbitrary objects with residual physics,” *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1307–1319, 2020.
- [7] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson, “Implicit behavioral cloning,” in *Conference on robot learning*. PMLR, 2022, pp. 158–168.
- [8] N. M. Shafiqullah, Z. Cui, A. A. Altanzaya, and L. Pinto, “Behavior transformers: Cloning  $k$  modes with one stone,” *Advances in neural information processing systems*, vol. 35, pp. 22955–22968, 2022.
- [9] 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*, p. 02783649241273668, 2023.
- [10] M. Shridhar, L. Manuelli, and D. Fox, “Perceiver-actor: A multi-task transformer for robotic manipulation,” in *Conference on Robot Learning*. PMLR, 2023, pp. 785–799.
- [11] T. Gervet, Z. Xian, N. Gkanatsios, and K. Fragkiadaki, “Act3d: 3d feature field transformers for multi-task robotic manipulation,” *arXiv preprint arXiv:2306.17817*, 2023.
- [12] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki, “3d diffuser actor: Policy diffusion with 3d scene representations,” in *8th Annual Conference on Robot Learning*, 2024.
- [13] A. Goyal, J. Xu, Y. Guo, V. Blukis, Y.-W. Chao, and D. Fox, “Rvt: Robotic view transformer for 3d object manipulation,” in *Conference on Robot Learning*. PMLR, 2023, pp. 694–710.
- [14] A. Goyal, V. Blukis, J. Xu, Y. Guo, Y.-W. Chao, and D. Fox, “Rvt-2: Learning precise manipulation from few demonstrations,” in *RSS 2024 Workshop: Data Generation for Robotics*.
- [15] Q. Lv, H. Li, X. Deng, R. Shao, Y. Li, J. Hao, L. Gao, M. Y. Wang, and L. Nie, “Spatial-temporal graph diffusion policy with kinematic modeling for bimanual robotic manipulation,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 17394–17404.
- [16] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, “3d diffusion policy,” *arXiv e-prints*, pp. arXiv–2403, 2024.
- [17] X. Chen, J. Djolonga, P. Padlewski, B. Mustafa, S. Changpinyo, J. Wu, C. R. Ruiz, S. Goodman, X. Wang, Y. Tay *et al.*, “Pali-x: On scaling up a multilingual vision and language model,” *arXiv preprint arXiv:2305.18565*, 2023.
- [18] H. Liu, C. Li, Q. Wu, and Y. J. Lee, “Visual instruction tuning,” *Advances in neural information processing systems*, vol. 36, pp. 34892–34916, 2023.
- [19] L. Beyer, A. Steiner, A. S. Pinto, A. Kolesnikov, X. Wang, D. Salz, M. Neumann, I. Alabdulmohsin, M. Tschanen, E. Bugliarello *et al.*, “Paligemma: A versatile 3b vlm for transfer,” *arXiv preprint arXiv:2407.07726*, 2024.
- [20] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge *et al.*, “Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution,” *arXiv preprint arXiv:2409.12191*, 2024.
- [21] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi *et al.*, “Open-vla: An open-source vision-language-action model,” *arXiv preprint arXiv:2406.09246*, 2024.
- [22] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, “pi0: A vision-language-action flow model for general robot control,” *arXiv preprint arXiv:2410.24164*, 2024.
- [23] D. Ghosh, H. R. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, J. Luo *et al.*, “Octo: An open-source generalist robot policy,” in *Robotics: Science and Systems*, 2024.
- [24] J. Wen, Y. Zhu, M. Zhu, Z. Tang, J. Li, Z. Zhou, X. Liu, C. Shen, Y. Peng, and F. Feng, “Diffusionvla: Scaling robot foundation models via unified diffusion and autoregression,” in *Forty-second International Conference on Machine Learning*.
- [25] A. O’Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain *et al.*, “Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 6892–6903.
- [26] X. Li, P. Li, M. Liu, D. Wang, J. Liu, B. Kang, X. Ma, T. Kong, H. Zhang, and H. Liu, “Towards generalist robot policies: What matters in building vision-language-action models,” *arXiv preprint arXiv:2412.14058*, 2024.
- [27] D. Qu, H. Song, Q. Chen, Y. Yao, X. Ye, Y. Ding, Z. Wang, J. Gu, B. Zhao, D. Wang *et al.*, “Spatialvla: Exploring spatial representations for visual-language-action model,” *arXiv preprint arXiv:2501.15830*, 2025.
- [28] Y. Liu, T. Wang, X. Zhang, and J. Sun, “Petr: Position embedding transformation for multi-view 3d object detection,” in *European Conference on Computer Vision*, 2022, pp. 531–548.
- [29] X. Lin, T. Lin, L. Huang, H. Xie, and Z. Su, “Bip3d: Bridging 2d images and 3d perception for embodied intelligence,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 9007–9016.
- [30] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su *et al.*, “Grounding dino: Marrying dino with grounded pre-training for open-set object detection,” in *European Conference on Computer Vision*. Springer, 2024, pp. 38–55.
- [31] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang *et al.*, “Qwen2. 5-vl technical report,” *arXiv preprint arXiv:2502.13923*, 2025.
- [32] J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang *et al.*, “GR00t n1: An open foundation model for generalist humanoid robots,” *arXiv preprint arXiv:2503.14734*, 2025.
- [33] J. Ho, A. Jain, and P. Abbeel, “Denosing diffusion probabilistic models,” *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.
- [34] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, “Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 5775–5787, 2022.
- [35] T. Zhao, V. Kumar, S. Levine, and C. Finn, “Learning fine-grained bimanual manipulation with low-cost hardware,” *Robotics: Science and Systems XIX*, 2023.