

Bridging Scale Discrepancies in Robotic Control via Language-Based Action Representations

Yuchi Zhang¹, Churui Sun¹, Shiqi Liang¹, Diyuan Liu², Chao Ji², Wei-Nan Zhang^{1*}, Ting Liu¹

¹Research Center for Social Computing and Interactive Robotics, Harbin Institute of Technology, Harbin, China

²State Key Laboratory of Cognitive Intelligence, iFLYTEK Research, China

{yczhang, crsun, sqliang, wnzhang, tliu}@ir.hit.edu.cn

{dyliu2, chaoji}@iflytek.com

Abstract

Recent end-to-end robotic manipulation research increasingly adopts architectures inspired by large language models to enable robust manipulation. However, a critical challenge arises from severe distribution shifts between robotic action data, primarily due to substantial numerical variations in action commands across diverse robotic platforms and tasks, hindering the effective transfer of pretrained knowledge. To address this limitation, we propose a semantically grounded linguistic representation to normalize actions for efficient pre-training. Unlike conventional discretized action representations that are sensitive to numerical scales, the motion representation specifically disregards numeric scale effects, emphasizing directionality instead. This abstraction mitigates distribution shifts, yielding a more generalizable pretraining representation. Moreover, using the motion representation narrows the feature distance between action tokens and standard vocabulary tokens, mitigating modality gaps. Multi-task experiments on two benchmarks demonstrate that the proposed method significantly improves generalization performance and transferability in robotic manipulation tasks.

Introduction

Recent advances in artificial intelligence have enabled models to acquire extensive knowledge from large-scale data, demonstrating robust generalization across tasks with minimal fine-tuning (Radford et al. 2019; Brown et al. 2020). Building on foundational robotics research that integrated vision and control (Chaumette and Hutchinson 2006; Saxena, Driemeyer, and Ng 2008), modern approaches increasingly adapt these large-scale AI methods to robotics (Yifan et al. 2025). In particular, language-conditioned action learning leverages both visual and linguistic inputs to guide robot actions, enabling more flexible and versatile manipulation capabilities. Large-scale datasets like Open X-Embodiment (OXE) (O’Neill et al. 2024) aggregate multi-modal robotic demonstrations across 22 robot embodiments and over 1 million tasks. By unifying visual (RGB/depth), proprioceptive, and language inputs with action trajectories in a standardized format, OXE facilitates cross-robot policy learning. Advanced works like OpenVLA, Octo, Pi_0 , and

RDT (Kim et al. 2024; Team et al. 2024; Black et al. 2024; Liu et al. 2024) leverage these datasets to explore model architectures and improving robotic manipulation.

Despite the successes in NLP through autoregressive pre-training, robotics models still face significant challenges in achieving similar transferable generalization. One of the main obstacles is distribution shifts across datasets caused by variations in collection environments, visual conditions, robot hardware, and data collection protocols. Consequently, current models frequently require extensive fine-tuning to perform satisfactorily in new domains. Additionally, existing language-conditioned imitation learning approaches often provide dynamic visual inputs at each timestep but maintain static language instructions. This imbalance limits the influence of language modality in guiding action generation, failing to fully utilize language’s potential.

To address the above limitations, we introduce a language-based intermediate representation to guide robot actions before execution, achieved through a rule-based mapping that transforms end-effector actions into coarse-grained language descriptions as alignment targets. To handle distribution shifts and ensure adaptability across diverse datasets, the proposed method incorporates a generalized motion generation method that applies spatial normalization and dynamic threshold adjustments. Leveraging the semantic richness and robust generalization of natural language, our unified pretraining strategy autonomously generates accurate, cost-efficient language alignment targets from diverse datasets without relying on external modules or manual intervention, thereby enhancing generalization and adaptability. We first train on a subset of the OXE dataset to capture execution patterns across varied environments, then fine-tune the model on manipulation benchmark datasets under language-alignment constraints to ensure semantic correspondence between actions and language. Experiments demonstrate that this approach enhances transferability, execution accuracy, and stability, with language alignment further bolstering robustness under diverse task conditions.

In summary, our contributions include:

- We propose a novel pretraining strategy leveraging rule-based linguistic representations to align action-language distributions across datasets, inherently capturing generalized motion-language relationships without manual annotations or external corrections, thus enhancing model gen-

*Corresponding author.

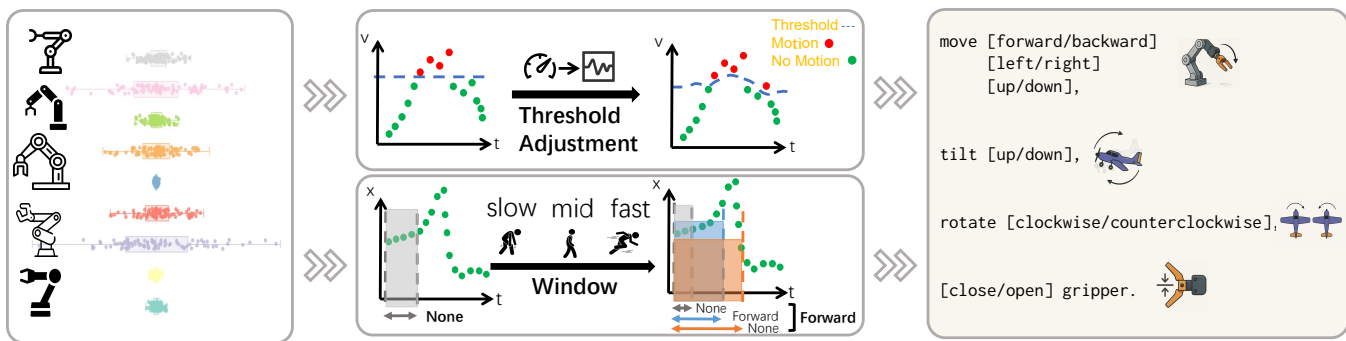


Figure 1: The proposed motion data generation pipeline. The left part illustrates the distributions of specific execution actions across different types of datasets; The middle part presents our threshold- and window-based detection framework along with its proposed improvements; The right part depicts the structure and representation of the generated motion outputs.

eralization and transferability.

- We propose an adaptive multi-scale motion detection method that dynamically adjusts thresholds and employs hierarchical windows, effectively suppressing motion jitter and false segmentation across datasets, significantly improving the accuracy of complex action recognition.
- Extensive evaluations on LIBERO (Liu et al. 2023) and Simpler Env (Li et al. 2024b) benchmarks validate our method’s superior accuracy, stability, and robustness compared to existing baselines.

Related Work

End-to-End Action Generation

In the manipulation field, there are many attempts to train models on a large scale end-to-end. Typical examples include RT1 (Brohan et al. 2022) and RT2 (Brohan et al. 2023), which use FiLM (Perez et al. 2018) and CLIP (Radford et al. 2021) for image encoding, transformer as the backbone, and discrete action space instead of act token for action decoding. RT2 (Brohan et al. 2023) also uses large-scale mixed data to allow the model to perform actions while retaining some multimodal QA knowledge. Other work from the same period includes Octo (Team et al. 2024), which also uses Transformer as the backbone and is pretrained on the largest robot manipulation dataset Open X-Embodiment. Moreover, there are many similar works, including Openvla (Kim et al. 2024) and P_{i_0} (Black et al. 2024). Some work has noticed that when using multiple robot arm data for pre-training, there will be issues of embodiment inconsistency. To solve this problem, RDT (Liu et al. 2024) introduces a Physically Interpretable Unified Action Space to unify data from different sources, while HPT (Wang et al. 2024) utilizes embodiment-specific tokenizers (“stems”), mapping the proprioception and visual sensing information of different robotic arms into a shared latent space.

Action Generation Assisted by Textual Guidance

Some researchers believe that in manipulation tasks, semantic expressions are becoming more diverse, making the mapping from high-level tasks to specific operations more dif-

ficult. To address this issue, some research proposes letting the model first learn the mapping from tasks to general language descriptions and then further learn specific operational actions. However, the model may have biases when generating actions based on language descriptions. Therefore, RT-H (Belkhal et al. 2024) introduced a manual intervention mechanism to correct errors in language descriptions, while ECoT (Zawalski et al. 2024) extended the language reasoning chain to guide correct action descriptions, exploring the effectiveness of ChatGPT in correcting actions. Additionally, Emma (Sun et al. 2024) further improved chain-of-thought generation and introduced explicit state information from trajectories as input to enhance the model’s task understanding and execution capabilities. Similarly, CoA (Li et al. 2024a) proposed Chain-of-Affordance, using the location of affordances in images as a chain of thought to guide the model in generating more robust actions. Meanwhile, some work (Qi et al. 2025) considers object orientations to be a key requirement for fine-grained manipulations tasks; They constructed a dataset of object-text-orientation pairs, excelling in many embodied tasks. In contrast to the above methods that rely on explicit guidance, we propose enhancing the model’s action generation capability through multi-dataset pretraining. This approach enables the model to produce more robust motion language descriptions, thereby significantly improving its generalization performance across tasks and datasets—without requiring additional reasoning chains or manual correction. (Fu et al. 2024).

Methods

This section presents our method in the order of action tokenization, motion generation, and model training. First, *Action Tokenizer* discretizes continuous action signals into token sequences to establish a learnable output space. Second, *Motion Generation* generates robust natural-language motion signals using adaptive thresholds and hierarchical temporal windows, serving as high-level semantic guidance. Finally, the *Two-Stage Training* stage employs a two-stage conditional generation strategy to progressively predict concrete actions from observations and instructions.

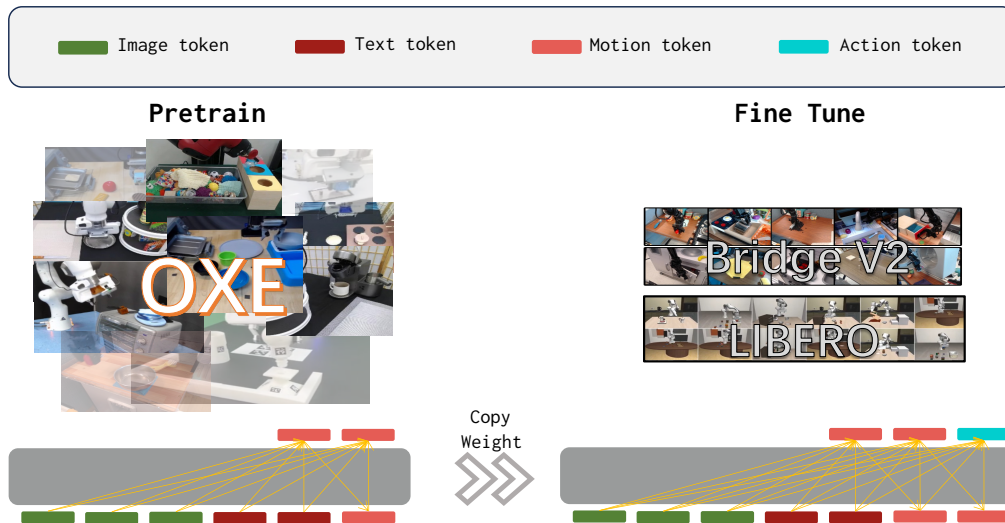


Figure 2: Two-stage training on Qwen2.5 (0.5B, 1.5B, 3B): pretraining predicts motion tokens; fine-tuning predicts motion then action tokens. Image tokens denote the observed visual input; text tokens denote the task instruction; motion tokens denote our proposed motion language; and action tokens denote the discrete action representation.

Action Tokenizer

Our action decoding method follows the approach of RT-2 and OpenVLA. Based on the task instruction and current observation input, the VLA should predict 7 action tokens consecutively, representing a 7-dimensional action the robot should execute, with each dimension corresponding to $(\Delta X, \Delta Y, \Delta Z, \Delta \text{roll}, \Delta \text{pitch}, \Delta \text{yaw}, \text{GripperState})$. The variables are normalized during training, and the output is denormalized during inference. Each normalized variable is discretized into 256 bins, where each bin is represented as a unique token. This transforms the action prediction task into a token-based sequence prediction task. In our VLA design, we appended 256 additional tokens to the tail, specifically to represent the 256 action tokens, denoted as $\langle \text{extra}_0 \rangle \dots \langle \text{extra}_{255} \rangle$. For normalizing, we exclude the outliers in each of the seven dimensions that fall outside the 1st and 99th percentile range. If outliers are included, the normalization range expands significantly, resulting in coarse-grained predictions and larger bin sizes, which can negatively impact precision.

Motion Generation

Previous work typically relies on manually defined thresholds and window sizes to generate motion signals. The **threshold** distinguishes actions by treating motion magnitude above it as an active movement and below it as no action in that dimension, while the **window** defines the temporal span over which motion displacements are accumulated into a single motion token. While such methods perform reasonably well on individual datasets, their simplicity makes them poorly suited for handling complex motion patterns across multiple datasets.

To enable collaborative motion generation in a multi-dataset setting, the normalization method described in subsection *Action Tokenizer* is first applied before generating

motion signals. Building upon this, we account for the jittering phenomenon commonly observed in robotic arms operating in real-world environments by replacing fixed thresholds with adaptive ones. Additionally, to accommodate the diverse types of robotic arm movements, we replace the single fixed-size window with a hierarchical detection window.

Motion Representation The “motion” representations we generated are a fixed set of natural language descriptions structured as follows: move [forward/backward] [left/right] [up/down], tilt [up/down], rotate [clockwise/counterclockwise], [open/close] gripper. Specifically, the keyword move describes positional displacement of the actuator along coordinate axes, while tilt and rotate denote angular rotations of the actuator, and gripper refers to its open-close actions. Each “motion” representation is composed of a combination of these three movement types. In cases where no movement is detected across all dimensions, the motion token is labeled as “stop”.

Threshold The threshold defines the minimum motion magnitude required to consider a change as meaningful—motion below this threshold is treated as no action in the corresponding dimension. To achieve more precise threshold determination for motion detection, we specifically take into account the jittering phenomena that may arise from faster movement speeds. We introduce a speed-based correction method, which adjusts the threshold to compensate for jitter caused by high-speed motion.

Let T_{base}^i denotes the basic threshold, and β represents the sensitivity coefficient. Additionally, τ represents the threshold adjustment window. The detecting formula is shown below, where s indexes the time steps within the window:

$$T_i(t) = T_{base}^i + \beta \cdot \frac{1}{\tau} \sum_{t-\tau}^t |\hat{\Delta}_i(s)| \quad (1)$$

Window Since motion generation often involves accumulation over multiple frames, the window determines the temporal span over which this accumulation occurs. Oriented to fixed window for motion detecting, incorporating the approach of fast-varying subsystem and slow-varying subsystem in singular systems, we designed the layered detecting window for motion categories comprising three temporal resolutions: fast, mid, and slow. Let $p(t) \in \mathbb{R}^3$ denote the 3D gripper position at time t , and let $T \in \mathbb{R}^+$ be a predefined movement threshold. To simplify expressions, we define the following shorthand notations:

- $\Delta_t p := p(t) - p(t - 1)$: unit-step displacement at time t
- $\Delta_{t_X} p := p(t) - p(t - \Delta t_X)$: displacement over a window of size Δt_X , where $X \in \{\text{fast, mid, slow}\}$

For brevity, we use subscripts f, m, s to denote fast, mid, slow respectively in the following definitions. The motion detectors for each temporal level are then defined as follows:

$$M_f := \|\Delta_{t_f} p\| > 2T \quad (2)$$

Focusing on regular movements, we modified the motion judging logic based on ECOT(Zawalski et al. 2024), ensuring the robotic arm is always in a moving process.

$$M_m := \|\Delta_{t_m} p\| > T \wedge \min_{\tau \in [t - \Delta t_m, t]} \|\Delta_{t_\tau} p\| > 0 \quad (3)$$

Focusing on the slow reactions in slow-varying systems, considering that these reactions usually have long response times and slow-changing state variables, we designed a threshold detection under a large window. Meanwhile, to avoid multiple segments of motion being recognized as a whole slow reaction due to an overly large window, we stipulated that the motion must proceed steadily in the same direction.

$$M_s := \|\Delta_{t_s} p\| > T \wedge \min_{\tau \in [t - \Delta t_s, t]} \|\Delta_{t_\tau} p\| > \frac{T}{2\Delta t_s} \quad (4)$$

We comprehensively evaluate the actions, identifying them only after each action detection has been passed.

$$\text{Motion}(t) := M_f(t) \vee M_m(t) \vee M_s(t) \quad (5)$$

Design Justification To compare the advantages and disadvantages of our proposed method with the fixed threshold-based approach used in ECoT, we manually annotated 5% of the data (or 3% for some larger datasets) and computed the average accuracy of action annotations. Given the distributional differences across datasets, we used separately tuned thresholds for each dataset when applying the baseline method.

The average annotation accuracy of our method reached 86.37%, significantly outperforming the ECoT-style threshold method, which achieved only 57.62%.

Upon further analysis of the failure cases, we observed that the threshold-based method—due to its simplicity—tends to falsely identify minor jitter during execution as multiple distinct actions. In contrast, our method effectively suppresses such false positives, leading to more stable and robust action recognition across different datasets.

Two-Stage Training

For each manipulation trajectory i , we associate a task instruction, formulated as: "What action should the robot take to {instruction}?". A trajectory consists of a sequence of discrete actions $A_i = (a_i^0, a_i^1, \dots, a_i^T)$, where T denotes the total number of steps in the trajectory. These actions align with video frames captured from a certain viewpoint, denoted $O_i = (o_i^0, o_i^1, \dots, o_i^T)$. Here, we employ third-person video frames as the image input and fix the temporal window to 1, leading each trajectory to produce one data instance per step. To enable an end-to-end cross-modal mapping, we introduce a language-based "motion" modality $M_i = (m_i^0, m_i^1, \dots, m_i^T)$ in the output to describe the actions. These coarse-grained labels serve as an intermediate representation, ultimately forming tuples of the form $(O_i^j, p_i, M_i^j, A_i^j)$, where i indexes the trajectory, j indexes the step within trajectory i , and p_i denotes the task instruction associated with the i -th trajectory. The strategy we aim to learn consists of two stages: firstly, $\phi_h(m|o, p)$ indicates that, conditioned on the current observation and instruction, the model generates motion tokens describing the upcoming actions in an autoregressive, next-token-prediction manner. Subsequently, $\phi_l(a|o, p, m)$ leverages these predicted motion tokens as contextual information to infer specific action tokens.

$$\phi(a, m|o, p) = \phi_h(m|o, p)\phi_l(a|o, p, m) \quad (6)$$

Motion-Only Pretraining We believe that the VLA model during pretraining encounters multi-source data with significantly different distributions, where action offsets are particularly severe, caused by the sampling frequencies and entity machine differences at the time of dataset construction. Therefore, existing pretraining often struggles to capture general information. In contrast, the motions generated on each dataset, as described in the previous subsection, are relatively more unified. Following the idea similar to curriculum learning, we start with easier tasks, so we hope that the pretraining phase can more efficiently capture general directional knowledge. Meanwhile, the difficulty of both learning and transfer will be greatly reduced, aligning with the principles of curriculum learning (Qi et al. 2024).

The format of the training data follows the construction method of VLM supervised fine-tuning data (llava), as presented in Table 1.

Downstream Fine-Tuning After a broad and diverse pretraining phase, we expect the model to acquire more general motion representations. However, due to data distribution shifts, none of the existing VLA models can achieve zero-shot generalization and need further adaptation through imitation learning data from downstream scenarios. Therefore, fine-tuning is an essential process.

Meanwhile, since our pretraining objective is to learn more general representations, although this motion also represents an action, it is too coarse-grained. As pointed out by (Li et al. 2024c), coarse-grained predictions are easier to make under the same conditions but perform far worse in execution compared to fine-grained ones. Therefore, we need

```

<START>system\n  $X_{\text{system}}$  <STOP> \n
<START>user\n What action should the robot take to  $X_{\text{instruct}}$  ? <STOP> \n
<START>motion\n  $X_{\text{motion}}$  <STOP> \n

```

Table 1: The input sequence used to pretrain the model. In practice. In our current implementation, $X_{\text{system}} = \text{You are Qwen, created by Alibaba Cloud. You are a helpful assistant.}$ and $\text{<START>} = \text{<|im_start|>}$, $\text{<STOP>} = \text{<|im_end|>}$. The model is trained to predict the robot motion and where to stop, and thus only **green sequence/tokens** are used to compute the loss in the auto-regressive model.

```

<START>system\n  $X_{\text{system}}$  <STOP> \n
<START>user\n What action should the robot take to  $X_{\text{instruct}}$  ? <STOP> \n
<START>motion\n  $X_{\text{motion}}$  <STOP> \n
<START>assistant\n  $X_{\text{action}}$  <STOP> \n

```

Table 2: Most settings are the same as in the previous subsection, but in this period, the model needs to predict both motion and robotic arm actions simultaneously.

finer-grained action tokens to represent the actions to be executed. The training data follows the construction format of VLM supervised fine-tuning data, as presented in Table 2.

Experimental Setup

Research Questions

In our experimental evaluation, we aim to answer the following questions:

- (RQ1) What is the individual contribution of each refinement on performance?
- (RQ2) Does the model with these refinements outperform established baselines and state-of-the-art approaches?
- (RQ3) Does adding a language output objective reduce the gap between action tokens and language tokens?

Baseline Methods

We compare with recent baselines including **Diffusion Policy** (Chi et al. 2023), **ScaleDP** (Zhu et al. 2024), **Octo** (Team et al. 2024), **OpenVLA** (Kim et al. 2024), **RT-1-x** (Brohan et al. 2022), and **ECoT** (Zawalski et al. 2024), covering diffusion-based, vision-language-action, and transformer-based policies. Details are provided in Appendix A.

Dataset

During pretraining, we utilized Open X-Embodiment, a large and diverse dataset containing hundreds of thousands of demonstrations. To reduce the computational cost during the pretraining phase, we selected 7 sub-datasets from it, including furniture-bench(Heo et al. 2023) and jaco(Dass et al. 2023), totalling approximately 12,000 trajectories. More details are in the Appendix B. This amount of data has shown promising results in demonstrating the benefits of motion pre-training in our experiments. To evaluate the generalization ability of the pretraining method, our pretraining dataset

does not include LIBERO and Bridge V2(Walke et al. 2024). Based on this, we generated motion data for pretraining using the pipeline introduced in Section Method.

For fine-tuning, we applied the same pipeline to LIBERO and Bridge V2 datasets. LIBERO features 130+ language-conditioned manipulation tasks for studying knowledge transfer in lifelong learning. Bridge V2 includes 7,200 demonstrations spanning 10 environments and 71 tasks in household scenarios.

Evaluation

LIBERO is a benchmark of 130+ language-conditioned tasks for Lifelong Decision-Making Learning (LLDM), focusing on knowledge transfer and skill personalization. We tested on four suites: Spatial, Goal, Object, and Long, following the open-source OpenVLA settings.

SimplerEnv offers a scalable simulation for real-world robot manipulation. For Bridge v2, SimplerEnv evaluates success rates on four tasks: Place a spoon on a towel, Place a carrot on a plate, Stack a green block on a yellow block and Place an eggplant in a yellow basket.

Implementation Details

Model Architecture Our architecture builds on OpenVLA, standardizing image resolution to 224×224 px and encoding with SigLIP(Zhai et al. 2023) and DINO v2(Oquab et al. 2024), followed by channel-wise concatenation. The LLM backbone uses Qwen2.5(Qwen et al. 2025) in three sizes: 0.5B, 1.5B, and 3B, with 256 special tokens added to the action tokenizer for 256 bins.

We replicated two-stage VLM training for Qwen2.5 using the LLaVA 1.5 data mixture in the Prismatic(Karamcheti et al. 2024) framework, then trained using our two-stage method. To verify the impact of pretraining, we conducted experiments from scratch, performing direct fine-tuning.

Fine-tuning Hyperparameters For pretraining, we used a batch size of 2048, and for fine-tuning, 512, with a learning

From Scratch		LIBERO-Spatial	LIBERO-Object	LIBERO-Goal	LIBERO-Long	Average	Δ Avg.
0.5B	w/ motion	84.0±0.9	86.6±0.9	78.0±1.1	46.0±1.3	73.7±0.6	+2.5
	w/o motion	84.8±0.9	85.8±0.9	69.6±1.2	44.4±1.3	71.2±0.6	
1.5B	w/ motion	84.6±0.9	85.2±0.9	76.4±1.1	47.8±1.3	73.5±0.6	+1.8
	w/o motion	84.0±0.9	84.8±1.0	71.0±1.2	46.8±1.3	71.7±0.6	
3B	w/ motion	83.6±0.9	86.8±0.9	79.6±1.0	48.0±1.3	74.5±0.6	+2.5
	w/o motion	85.0±0.9	86.4±0.9	71.2±1.1	45.4±1.3	72.0±0.6	

Table 3: Success rates (%) on LIBERO benchmark (models trained from scratch)

From Scratch		Spoon/Towel	Carrot/Plate	Stack Blocks	Eggplant/Basket	Average	Δ Avg.
0.5B	w/ motion	16.7	19.6	0	16.4	13.2	+1.5
	w/o motion	14.1	20.5	0	12.2	11.7	
1.5B	w/ motion	22.7	16.5	0	20.9	15.0	-0.3
	w/o motion	33.1	19.8	0	8.6	15.3	
3B	w/ motion	35.8	25.1	0	56.5	29.4	+9.6
	w/o motion	28.5	24.4	0	26.3	19.8	

Table 4: Success rates (%) on SimplerEnv benchmark (models trained from scratch)

rate of $2e-5$. All experiments were conducted on A100-80G GPUs. For detailed information regarding the training duration, inference time, and computational efficiency, as well as the time spent on each experimental phase and reasoning process, please refer to Appendix C.

Experimental Results and Analysis

How does each refinement impact performance individually? (RQ1)

To validate the effectiveness of our proposed pretraining strategy compared to other existing methods, we conducted comprehensive experiments across different model sizes. Specifically, we pretrained models solely focusing on motion generation and models focusing exclusively on action generation. Subsequent fine-tuning was performed across various benchmarks to evaluate their performance. Experimental results clearly demonstrate that models pretrained with our motion-focused method achieve significantly higher success rates (SR). Additionally, by comparing results from Table 3–6, it becomes evident that pretraining with motion provides a substantially greater improvement over baseline models trained from scratch. This further underscores the effectiveness and importance of incorporating motion learning during the pretraining stage. However, it was observed that the 1.5B parameter model exhibited limited performance gains in the SimplerEnv benchmark. We suggest this phenomenon arises primarily due to the gap between fine-tuning data, which was collected from real-world scenarios, and the simulated testing environment, compounded by the constraints imposed by the limited pa-

After Pretrain		LIBERO-Spatial	LIBERO-Object	LIBERO-Goal	LIBERO-Long	Average	Δ Avg.
0.5B	w/ motion	86.0±0.9	84.8±0.9	76.4±1.1	51.2±1.3	74.6±0.6	+3.2
	w/o motion	85.2±0.9	82.2±1.0	69.0±1.2	49.0±1.3	71.4±0.6	
1.5B	w/ motion	86.8±0.9	84.8±0.9	75.2±1.1	51.6±1.3	74.6±0.6	+3.0
	w/o motion	84.4±0.9	82.8±1.0	69.8±1.2	49.4±1.3	71.6±0.6	
3B	w/ motion	84.8±0.9	90.0±0.8	82.2±1.0	55.4±1.3	78.1±0.5	+6.9
	w/o motion	82.0±1.0	85.6±0.9	70.2±1.2	46.8±1.3	71.2±0.6	

Table 5: Success rates (%) on LIBERO benchmark (models trained based on pretraining)

After Pretrain		Spoon/Towel	Carrot/Plate	Stack Blocks	Eggplant/Basket	Average	Δ Avg.
0.5B	w/ motion	21.1	24.4	0.0	10.7	14.1	+2.3
	w/o motion	18.5	12.4	0.0	16.3	11.8	
1.5B	w/ motion	30.8	20.8	0.0	24.4	19.0	+0.8
	w/o motion	34.6	22.1	0.0	16.2	18.2	
3B	w/ motion	44.0	36.2	0.0	61.1	35.3	+14.1
	w/o motion	26.4	28.4	0.0	30.1	21.2	

Table 6: Success rates (%) on SimplerEnv benchmark (models trained based on pretraining)

parameter scale.

Furthermore, we aimed to assess the efficacy of our two proposed methods for optimizing motion generation quality: adjusting the window size and threshold parameters. Experiments were conducted using the 0.5B parameter model on the LIBERO benchmark, comparing three scenarios—without motion pretraining, with original motion pretraining, and with our optimized motion pretraining. Results from Table 7 align well with human judgment assessments described in our methodology, demonstrating that our optimization techniques substantially enhance motion generation quality. This robust evidence confirms the significant contribution of our optimization strategies in improving pretraining outcomes.

Does our refined model outperform baselines and SOTA? (RQ2)

We compared our method with state-of-the-art manipulation baselines. Unlike these baselines, our objective explicitly aligns motion components—the aspect most sensitive to dataset-driven numeric shifts. This simplification accelerates

Method	LIBERO-Spatial	LIBERO-Object	LIBERO-Goal	LIBERO-Long	Average
Ours (0.5B)	86.2±0.9	84.6±0.9	76.2±1.1	51.1±1.3	74.5±0.6
with raw motion	86.0±0.9	83.2±1.0	74.1±1.1	50.5±1.3	73.5±0.6
w/o motion	85.1±0.9	82.3±1.0	69.0±1.2	49.3±1.3	71.4±0.6

Table 7: Success rates (%) on Ours and raw motion and without motion

Method	LIBERO-Spatial	LIBERO-Object	LIBERO-Goal	LIBERO-Long	Average	
Diffusion Policy	78.3±1.1	92.5±0.7	68.3±1.2	50.5±1.3	72.4±0.7	
ScaleDP	79.1±0.7	90.4±0.9	73.6±0.8	48.4±1.2	72.9±0.5	
Octo	78.9±1.0	85.7±0.9	84.6±0.9	51.1±1.3	75.1±0.6	
Openvla	84.7±0.9	88.4±0.8	79.2±1.0	53.7±1.3	76.5±0.6	
Ours	0.5B	86.0±0.9	84.8±0.9	76.4±1.1	51.2±1.3	74.6±0.6
	1.5B	86.8±0.9	84.8±0.9	75.2±1.1	51.6±1.3	74.6±0.6
	3B	84.8±0.9	90.0±0.8	82.2±1.0	55.4±1.3	78.1±0.5

Table 8: Success rates (%) on LIBERO benchmark (models trained based on pretraining)

Method	Spoon / Towel	Carrot / Plate	Stack Blocks	Eggplant/ Basket	Average	
RT1-x	0.0	4.2	0.0	0.0	1.1	
Octo-Base	15.8	12.5	0.0	41.7	17.5	
Octo-Small	41.7	8.2	0.0	56.7	26.7	
Openvla	4.2	0.0	0.0	12.5	4.2	
ECoT	40.2	11.7	0.0	28.4	20.1	
Ours	0.5B	21.1	24.4	0.0	10.7	14.1
	1.5B	30.8	20.8	0.0	24.4	19.0
	3B	44.0	36.2	0.0	61.1	35.3

Table 9: Success rates (%) on SimplerEnv benchmark (models trained based on pretraining)

ates convergence and yields superior results on two benchmarks, proving the approach captures transferable motion directions with modest compute.

Specifically, we fine-tuned the pretrained models and evaluated them on the LIBERO and SimplerEnv benchmarks. As shown in Table 8 and Table 9, our approach consistently surpasses baselines lacking motion tokens, and it also outperforms models trained entirely from scratch. We further compared ECoT (7B), trained on the same Bridge dataset, and found our method still attained superior performance. Notably, the OpenVLA (7B) variant of our approach delivered particularly competitive results while requiring fewer pretraining data and smaller model sizes.

All models failed to succeed in the task “stack green block on yellow block” in SimplerEnv, This may be because it requires a higher level of precision than other tasks, because it first needs to grab a very small block and put it accurately on another very small block. Of course, we think the main reason is that the training data used is Bridge v2, which is collected from the real world, and our test environment is SimplerEnv, which is a test environment replicated in simulation based on the Bridge v2 dataset, which is different from the actual environment.

Does a language output objective reduce action-language distance? (RQ3)

Building on the findings from RQ2, we further investigate the effect of incorporating motion tokens on representation

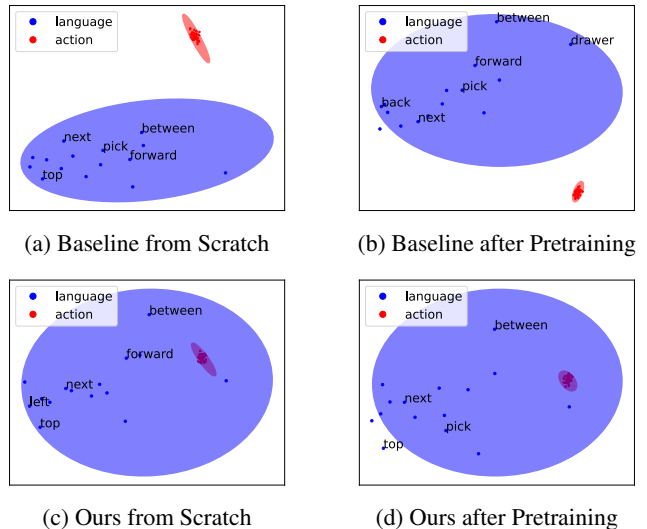


Figure 3: Comparison of four different experimental setups

alignment. Specifically, we leverage PCA and confidence ellipses to visualize embeddings from the spatial task of the LIBERO benchmark (see Fig. 3). We selected one of the train-sets on the spatial task, extracted the embeddings of the model under four training conditions (w&w/o pretraining; w&w/o motion), and then took the vectors corresponding to action tokens and motion tokens in the vocabulary from the embeddings, used principal component analysis to reduce the dimensionality, and then drew its confidence ellipse and marked the relevant tokens at the corresponding points.

Visualizations indicate that in end-to-end models trained for action token generation, the token features deviate significantly from those of the original vocabulary. Incorporating our motion representation (whether pretrained or trained from scratch) reduces this gap, helping to bridge the modality difference (Wei et al. 2025) and leading to more efficient training. Moreover, pretraining yields more clustered action token features, coinciding with improved manipulation performance, whereas training from scratch results in more dispersed representations, suggesting insufficient convergence.

Conclusion

In this paper, we introduce a novel pretraining strategy using language-modal action representations (motion) to tackle generalization issues caused by numerical distribution shifts across robotic platforms and tasks. Our method converts numerical actions into abstract directional semantic descriptions, significantly reducing distributional discrepancies and enabling efficient autonomous learning of generalized motion-language alignments. Experiments confirm that integrating motion tokens effectively bridges representational gaps between action and language modalities, enhancing generalization and transferability across various robotic manipulation benchmarks, all without external modules or manual intervention. Future work will focus on further optimizing the pretraining strategy to advance language-guided robotic manipulation toward practical applications.

Acknowledgments

The authors would like to thank all the anonymous reviewers for their insightful comments. Meanwhile, the authors would like to thank Prof. Wei-Nan Zhang and Dr. Yuanxing Liu for their help on revising the manuscript. This research was supported by the National Key Research and Development Program (No.2022YFF0902100), the National Natural Science Foundation of China (No. 92470205) and the National Natural Science Foundation of China (No. 20230320).

References

- Belkhale, S.; Ding, T.; Xiao, T.; Sermanet, P.; Vuong, Q.; Tompson, J.; Chebotar, Y.; Dwibedi, D.; and Sadigh, D. 2024. Rt-h: Action hierarchies using language. *arXiv preprint arXiv:2403.01823*.
- Black, K.; Brown, N.; Driess, D.; Esmail, A.; Equi, M.; Finn, C.; Fusai, N.; Groom, L.; Hausman, K.; Ichter, B.; et al. 2024. π 0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*.
- Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Chen, X.; Choromanski, K.; Ding, T.; Driess, D.; Dubey, A.; Finn, C.; et al. 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*.
- Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Dabis, J.; Finn, C.; Gopalakrishnan, K.; Hausman, K.; Herzog, A.; Hsu, J.; et al. 2022. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*.
- Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. *arXiv:2005.14165*.
- Chaumette, F.; and Hutchinson, S. 2006. Visual servo control. I. Basic approaches. *IEEE Robotics & Automation Magazine*, 13(4): 82–90.
- Chi, C.; Feng, S.; Du, Y.; Xu, Z.; Cousineau, E.; Burchfiel, B.; and Song, S. 2023. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*.
- Dass, S.; Yapeter, J.; Zhang, J.; Pertsch, K.; Nikolaidis, S.; and Lim, J. J. 2023. CLVR Jaco Play Dataset.
- Fu, D.; Qi, B.; Gao, Y.; Jiang, C.; Dong, G.; and Zhou, B. 2024. MSI-Agent: Incorporating Multi-Scale Insight into Embodied Agents for Superior Planning and Decision-Making. *arXiv preprint arXiv:2409.16686*.
- Heo, M.; Lee, Y.; Lee, D.; and Lim, J. J. 2023. Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation. *The International Journal of Robotics Research*, 02783649241304789.
- Karamcheti, S.; Nair, S.; Balakrishna, A.; Liang, P.; Kollar, T.; and Sadigh, D. 2024. Prismatic VLMs: Investigating the Design Space of Visually-Conditioned Language Models. *arXiv:2402.07865*.
- Kim, M. J.; Pertsch, K.; Karamcheti, S.; Xiao, T.; Balakrishna, A.; Nair, S.; Rafailov, R.; Foster, E.; Lam, G.; Sanketi, P.; et al. 2024. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*.
- Li, J.; Zhu, Y.; Tang, Z.; Wen, J.; Zhu, M.; Liu, X.; Li, C.; Cheng, R.; Peng, Y.; and Feng, F. 2024a. Improving Vision-Language-Action Models via Chain-of-Affordance. *arXiv preprint arXiv:2412.20451*.
- Li, X.; Hsu, K.; Gu, J.; Pertsch, K.; Mees, O.; Walke, H. R.; Fu, C.; Lunawat, I.; Sieh, I.; Kirmani, S.; Levine, S.; Wu, J.; Finn, C.; Su, H.; Vuong, Q.; and Xiao, T. 2024b. Evaluating Real-World Robot Manipulation Policies in Simulation. *arXiv preprint arXiv:2405.05941*.
- Li, X.; Li, P.; Liu, M.; Wang, D.; Liu, J.; Kang, B.; Ma, X.; Kong, T.; Zhang, H.; and Liu, H. 2024c. Towards generalist robot policies: What matters in building vision-language-action models. *arXiv preprint arXiv:2412.14058*.
- Liu, B.; Zhu, Y.; Gao, C.; Feng, Y.; Liu, Q.; Zhu, Y.; and Stone, P. 2023. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. *arXiv:2306.03310*.
- Liu, S.; Wu, L.; Li, B.; Tan, H.; Chen, H.; Wang, Z.; Xu, K.; Su, H.; and Zhu, J. 2024. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*.
- Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; Assran, M.; Ballas, N.; Galuba, W.; Howes, R.; Huang, P.-Y.; Li, S.-W.; Misra, I.; Rabbat, M.; Sharma, V.; Synnaeve, G.; Xu, H.; Jegou, H.; Mairal, J.; Labatut, P.; Joulin, A.; and Bojanowski, P. 2024. DINOv2: Learning Robust Visual Features without Supervision. *arXiv:2304.07193*.
- O’Neill, A.; Rehman, A.; Maddukuri, A.; Gupta, A.; Padalkar, A.; Lee, A.; Pooley, A.; Gupta, A.; Mandlekar, A.; Jain, A.; Tung, A.; Bewley, A.; Herzog, A.; Irpan, A.; Khazatsky, A.; Rai, A.; Gupta, A.; Wang, A.; Singh, A.; Garg, A.; Kembhavi, A.; Xie, A.; Brohan, A.; Raffin, A.; Sharma, A.; Yavary, A.; Jain, A.; Balakrishna, A.; Wahid, A.; Burgess-Limerick, B.; Kim, B.; Schölkopf, B.; Wulfe, B.; Ichter, B.; Lu, C.; Xu, C.; Le, C.; Finn, C.; Wang, C.; Xu, C.; Chi, C.; Huang, C.; Chan, C.; Agia, C.; Pan, C.; Fu, C.; Devin, C.; Xu, D.; Morton, D.; Driess, D.; Chen, D.; Pathak, D.; Shah, D.; Büchler, D.; Jayaraman, D.; Kalashnikov, D.; Sadigh, D.; Johns, E.; Foster, E.; Liu, F.; Ceola, F.; Xia, F.; Zhao, F.; Stulp, F.; Zhou, G.; Sukhatme, G. S.; Salhotra, G.; Yan, G.; Feng, G.; Schiavi, G.; Berseth, G.; Kahn, G.; Wang, G.; Su, H.; Fang, H.-S.; Shi, H.; Bao, H.; Ben Amor, H.; Christensen, H. I.; Furuta, H.; Walke, H.; Fang, H.; Ha, H.; Mordatch, I.; Radosavovic, I.; Leal, I.; Liang, J.; Abou-Chakra, J.; Kim, J.; Drake, J.; Peters, J.; Schneider, J.; Hsu, J.; Bohg, J.; Bingham, J.; Wu, J.; Gao, J.; Hu, J.; Wu, J.; Sun, J.; Luo, J.; Gu, J.; Tan, J.; Oh, J.; Wu, J.; Lu, J.; Yang, J.; Malik, J.; Silvério, J.; Hejna, J.; Booher, J.; Tompson, J.; Yang, J.; Salvador, J.; Lim, J. J.; Han, J.; Wang, K.; Rao, K.; Pertsch, K.; Hausman, K.; Go, K.; Gopalakrishnan, K.; Goldberg, K.; Byrne, K.; Oslund, K.; Kawaharazuka, K.; Black, K.; Lin, K.; Zhang, K.; Ehsani, K.; Lekkala, K.; Ellis,

- K.; Rana, K.; Srinivasan, K.; Fang, K.; Singh, K. P.; Zeng, K.-H.; Hatch, K.; Hsu, K.; Itti, L.; Chen, L. Y.; Pinto, L.; Fei-Fei, L.; Tan, L.; Fan, L. J.; Ott, L.; Lee, L.; Weihs, L.; Chen, M.; Lepert, M.; Memmel, M.; Tomizuka, M.; Itkina, M.; Castro, M. G.; Spero, M.; Du, M.; Ahn, M.; Yip, M. C.; Zhang, M.; Ding, M.; Heo, M.; Srirama, M. K.; Sharma, M.; Kim, M. J.; Kanazawa, N.; Hansen, N.; Heess, N.; Joshi, N. J.; Suenderhauf, N.; Liu, N.; Di Palo, N.; Shafiullah, N. M. M.; Mees, O.; Kroemer, O.; Bastani, O.; Sanketi, P. R.; Miller, P. T.; Yin, P.; Wohlhart, P.; Xu, P.; Fagan, P. D.; Mirano, P.; Sermanet, P.; Abbeel, P.; Sundaresan, P.; Chen, Q.; Vuong, Q.; Rafailov, R.; Tian, R.; Doshi, R.; Martín-Martín, R.; Bajjal, R.; Scalise, R.; Hendrix, R.; Lin, R.; Qian, R.; Zhang, R.; Mendonca, R.; Shah, R.; Hoque, R.; Julian, R.; Bustamante, S.; Kirmani, S.; Levine, S.; Lin, S.; Moore, S.; Bahl, S.; Dass, S.; Sonawani, S.; Song, S.; Xu, S.; Haldar, S.; Karamcheti, S.; Adebola, S.; Guist, S.; Nasiriany, S.; Schaal, S.; Welker, S.; Tian, S.; Ramamoorthy, S.; Dasari, S.; Belkhale, S.; Park, S.; Nair, S.; Mirchandani, S.; Osa, T.; Gupta, T.; Harada, T.; Matsushima, T.; Xiao, T.; Kollar, T.; Yu, T.; Ding, T.; Davchev, T.; Zhao, T. Z.; Armstrong, T.; Darrell, T.; Chung, T.; Jain, V.; Vanhoucke, V.; Zhan, W.; Zhou, W.; Burgard, W.; Chen, X.; Wang, X.; Zhu, X.; Geng, X.; Liu, X.; Liangwei, X.; Li, X.; Lu, Y.; Ma, Y. J.; Kim, Y.; Chebotar, Y.; Zhou, Y.; Zhu, Y.; Wu, Y.; Xu, Y.; Wang, Y.; Bisk, Y.; Cho, Y.; Lee, Y.; Cui, Y.; Cao, Y.; Wu, Y.-H.; Tang, Y.; Zhu, Y.; Zhang, Y.; Jiang, Y.; Li, Y.; Li, Y.; Iwasawa, Y.; Matsuo, Y.; Ma, Z.; Xu, Z.; Cui, Z. J.; Zhang, Z.; and Lin, Z. 2024. Open X-Embodiment: Robotic Learning Datasets and RT-X Models : Open X-Embodiment Collaboration0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 6892–6903.
- Perez, E.; Strub, F.; De Vries, H.; Dumoulin, V.; and Courville, A. 2018. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.
- Qi, B.; Chen, X.; Gao, J.; Li, D.; Liu, J.; Wu, L.; and Zhou, B. 2024. Interactive continual learning: Fast and slow thinking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12882–12892.
- Qi, Z.; Zhang, W.; Ding, Y.; Dong, R.; Yu, X.; Li, J.; Xu, L.; Li, B.; He, X.; Fan, G.; et al. 2025. SoFar: Language-Grounded Orientation Bridges Spatial Reasoning and Object Manipulation. *arXiv preprint arXiv:2502.13143*.
- Qwen, ;; Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; Lin, H.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Yang, J.; Zhou, J.; Lin, J.; Dang, K.; Lu, K.; Bao, K.; Yang, K.; Yu, L.; Li, M.; Xue, M.; Zhang, P.; Zhu, Q.; Men, R.; Lin, R.; Li, T.; Tang, T.; Xia, T.; Ren, X.; Ren, X.; Fan, Y.; Su, Y.; Zhang, Y.; Wan, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; and Qiu, Z. 2025. Qwen2.5 Technical Report. *arXiv:2412.15115*.
- Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PmlR.
- Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language Models are Unsupervised Multitask Learners.
- Saxena, A.; Driemeyer, J.; and Ng, A. Y. 2008. Robotic grasping of novel objects using vision. *The International Journal of Robotics Research*, 27(2): 157–173.
- Sun, Q.; Hong, P.; Pala, T. D.; Toh, V.; Tan, U.; Ghosal, D.; Poria, S.; et al. 2024. Emma-X: An Embodied Multimodal Action Model with Grounded Chain of Thought and Look-ahead Spatial Reasoning. *arXiv preprint arXiv:2412.11974*.
- Team, O. M.; Ghosh, D.; Walke, H.; Pertsch, K.; Black, K.; Mees, O.; Dasari, S.; Hejna, J.; Kreiman, T.; Xu, C.; et al. 2024. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*.
- Walke, H.; Black, K.; Lee, A.; Kim, M. J.; Du, M.; Zheng, C.; Zhao, T.; Hansen-Estruch, P.; Vuong, Q.; He, A.; Myers, V.; Fang, K.; Finn, C.; and Levine, S. 2024. BridgeData V2: A Dataset for Robot Learning at Scale. *arXiv:2308.12952*.
- Wang, L.; Chen, X.; Zhao, J.; and He, K. 2024. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. *Advances in Neural Information Processing Systems*, 37: 124420–124450.
- Wei, M.; Zhang, W.-N.; Zhang, C.; Ding, Y.; Di, D.; Ren, L.; Chen, W.; and Liu, T. 2025. PRISM: A Benchmark for Unveiling Cross-modal Knowledge Inconsistency in Large Vision-Language Models. In *Proceedings of the 33rd ACM International Conference on Multimedia*, 11121–11129.
- Yifan, C.; Wei, M.; Wang, X.; Liu, Y.; Wang, J.; Song, H.; Ma, L.; Di, D.; Sun, C.; Liu, K.; Qi, L.; Yu, J.; Tian, X.; Liang, S.; Duan, C.; Hong, Z.; Zhang, W.; and Liu, T. 2025. Embodied AI: A Survey on the Evolution from Perceptive to Behavioral Intelligence. *SmartBot*.
- Zawalski, M.; Chen, W.; Pertsch, K.; Mees, O.; Finn, C.; and Levine, S. 2024. Robotic control via embodied chain-of-thought reasoning. *arXiv preprint arXiv:2407.08693*.
- Zhai, X.; Mustafa, B.; Kolesnikov, A.; and Beyer, L. 2023. Sigmoid Loss for Language Image Pre-Training. *arXiv:2303.15343*.
- Zhu, M.; Zhu, Y.; Li, J.; Wen, J.; Xu, Z.; Liu, N.; Cheng, R.; Shen, C.; Peng, Y.; Feng, F.; and Tang, J. 2024. Scaling Diffusion Policy in Transformer to 1 Billion Parameters for Robotic Manipulation. *arXiv:2409.14411*.