LatBot: Distilling Universal Latent Actions for Vision-Language-Action Models

Zuolei Li[1,2†] Xingyu Gao[1,2✉] Xiaofan Wang[1,2†] Jianlong Fu[3]
[1] Institute of Microelectronics, Chinese Academy of Sciences
[2] University of Chinese Academy of Sciences [3] Microsoft Research
{lizuolei24, wangxiaofan24}@ime.ac.cn, gxy9910@gmail.com, jianf@microsoft.com
Project Page: mm-robot/LatBot

Abstract

Learning transferable latent actions from large-scale object manipulation videos can significantly enhance generalization in downstream robotics tasks, as such representations are agnostic to different robot embodiments. Existing approaches primarily rely on visual reconstruction objectives while neglecting physical priors, leading to suboptimal performance in learning universal representations. To address these challenges, we propose a Universal Latent Action Learning framework that takes task instructions and multiple frames as inputs, and optimizes both future frame reconstruction and action sequence prediction. Unlike prior works, incorporating action predictions (e.g., gripper or hand trajectories and orientations) allows the model to capture richer physical priors such as real-world distances and orientations, thereby enabling seamless transferability to downstream tasks. We further decompose the latent actions into learnable motion and scene tokens to distinguish the robot's active movements from environmental changes, thus filtering out irrelevant dynamics. By distilling the learned latent actions into the latest VLA models, we achieve strong performance across both simulated (SIMPLER and LIBERO) and real-world robot settings. Notably, with only 10 real-world trajectories per task collected on a Franka robot, our approach successfully completes all five challenging tasks, demonstrating strong few-shot transferability in robotic manipulation.

1. Introduction

Latent action learning has recently emerged as a promising research direction in the field of vision-language-action (VLA) models [5, 26, 34, 43, 51, 61, 65]. Its core idea is to extract and compress motion semantics between consecutive frames into compact latent representations that are agnostic to different robot embodiments. Unlike traditional methods [5, 26, 42] that rely on annotated action data, this paradigm enables models to leverage large-scale human videos, thereby significantly expanding the available training sources for robotic policies and overcoming the limitations of conventional robot datasets in terms of diversity and generalization.

However, as shown in Figure 1, existing latent action models (LAM) [8, 9, 12, 57] usually suffer from the following challenges. First, the absence of task instruction guidance prevents the latent action from capturing task-relevant changes (e.g., Genie [7]). Second, insufficient utilization of multiple frames results in imprecise latent action representations incapable of accurately capturing motion dynamics (e.g., UniVLA [9]). Third, the latent actions often focus on visual appearance changes but lacking physical awareness, causing a semantic gap between latent action representations and real executable actions. These limitations hinder the effective transfer of the learned latent actions to downstream tasks, as they fail to provide reliable cues for planning, limiting their ability to generalize from visual perception to real-world robotic execution.

To address these issues, we propose **LatBot**, a universal latent action learning framework for robotic tasks, which learns latent actions under the guidance of task instructions and multi-frame inputs, constrained by both visual and action generation objectives. This design enables VLA models to generalize more effectively across downstream tasks with few-shot samples (e.g., 25 demonstrations used in existing works [45]). First, we design two complementary learnable latent action tokens: *scene tokens* to capture passive environmental changes such as object position, pose, and background dynamics, and *motion tokens* to encode the robot's active movements such as end-effector translation, rotation, and gripper actions. This design explicitly disentangles environmental variations from robot-induced motion, leading to more structured latent actions that improve motion understanding and action prediction. Second, we

✉ Corresponding author: gxy9910@gmail.com
† Work conducted during internship at Microsoft Research

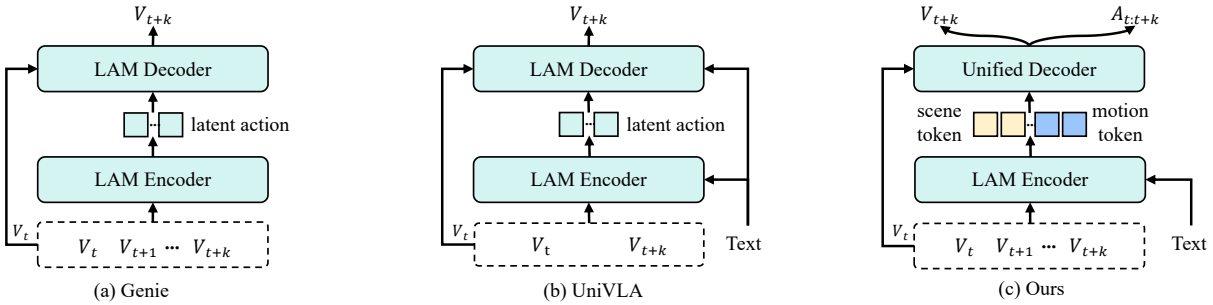

Figure 1. Different paradigms in latent action modeling (LAM). Existing methods often ignore disentangling robot actions from environmental changes. In contrast, we learn a disentangled representation and decode latent actions into both the future visual frame V_{t+k} and physical actions $A_{t:t+k}$ that enables more accurate and transferable control for downstream tasks.

propose a unified decoder that conditions on the latent actions to jointly guide future frame reconstruction and interframe action generation. It enables the model to learn universal latent actions that not only enhance the prediction of diverse real-world robotic manipulation scenes, but also bridge the gap with real actions, ultimately improving transferability to downstream manipulation tasks. To optimize latent action representations, we introduce bidirectional interactions between visual and action representations, where scene dynamics guide action generation and motion tokens refine visual reconstruction, enabling mutual reinforcement.

To effectively transfer the learned latent action knowledge into VLA models, we present an effective strategy for knowledge distillation. This strategy enables VLMs to inherit latent action knowledge while preserving its reasoning and instruction-understanding capabilities. It also allows the model to capture rich physical priors, ultimately enhancing generalization and transferability in real-world robotic manipulation. Specifically, we design two types of loss functions: *Latent Action Alignment Loss* and *Reasoning Preservation Loss*. The former transfers physical priors of latent actions from the teacher model (LAM) to the student (VLM) by aligning their latent action representations through both MSE and KL divergence. This helps the student learn physics-aware latent actions that can capture task-relevant motion patterns and future visual changes, allowing the model to rapidly adapt to new manipulation tasks with few shot samples. The latter adopts a next-token prediction objective, enabling the student to generate subtask descriptions based on the current frame and task instruction. It preserves the reasoning and instruction-following abilities of VLMs, ensuring that the distilled model remains robust and generalizable for complex robotic manipulation.

We pre-train the latent action model and perform knowledge distillation on diverse object manipulation datasets encompassing both robot and human hand demonstrations. These datasets include OXE [52], AgiBoT [8], and EgoDex [18] that represents the largest and most diverse publicly available collection of dexterous human manipulation data. The diversity of scenes and embodiments encourages the model to learn universal latent actions that generalize across visual domains and capture shared task patterns under different scenes, enhancing transferability to downstream robotic tasks. Compared with the latest VLA models [5, 20], our approach achieves superior performance in both simulation and real-world environments.

2. Related Work

2.1. Vision-Language-Action Model

Vision–Language–Action (VLA) models extend Vision–Language Models (VLMs) [1, 54] to generate robot actions conditioned on visual observations and language instructions. Early efforts such as RT-1 [6] and Octo [51] employ transformer-based policies trained on large-scale collections of robotic trajectories spanning diverse tasks, objects, and environments. RT-2 [65] further fine-tunes a pretrained VLM with both vision–language data and robotic demonstrations, discretizing actions into text-like tokens. Following a similar strategy, OpenVLA [26] adapts the Prismatic VLM [24] on the Open X-Embodiment dataset [41]. Other approaches integrate VLMs with specialized action modules. For instance, RoboFlamingo [28] appends a policy head for action prediction, while π_0 [5] leverages PaliGemma [3] for scene understanding and a flow-matching expert for continuous control. Furthermore, several methods also incorporate goal images [4] or video prediction [19] as auxiliary tasks to enhance planning and execution. Nevertheless, these methods rely heavily on interactive datasets with ground-truth action labels, which substantially limits the scalability and generalization of VLA models.

2.2. Latent Action Model

Recent research [9, 11, 57] has explored latent actions to address the scalability limitations of VLA models that

rely on ground-truth action labels. Latent actions provide compact and transferable representations, enabling learning from large-scale, unlabeled videos. Early studies such as Genie [7] and LAPO [47] introduced unsupervised latent action modeling in video game environments, while DynaMo [14] extended this idea with inverse and forward dynamics to learn structured state representations. In robotic learning, several methods [9, 11, 57, 58] incorporate latent actions into VLA pretraining, enabling policy learning without explicit action supervision. For example, LAPA [57] and ViLLA-X [11] extend latent action learning to both human and robot videos, facilitating cross-domain transfer between human demonstrations and robotic executions. Moto-GPT [12] focuses on motion-centric representation learning by converting videos into discrete motion tokens and co-finetuning them with real robot actions to bridge motion understanding and control. UniVLA [9] adopts a two-stage pipeline to learn task-centric latent actions, which shows promising results. However, existing approaches remain limited by sub-optimal latent representations. In contrast, we propose to explicitly disentangle latent actions into transferable components (including motion and scene tokens), and align them with real physical states (e.g., translation and rotation), which makes them easier to distill into downstream robotic tasks.

3. Approach

In this section, we introduce the proposed universal latent action learning framework, **LatBot**, which consists of two key components: Latent Action Disentanglement and Unified Decoder, which are jointly optimized during training. After pre-training the latent action representation, we further distill the learned motion knowledge into VLMs to enhance their action awareness while preserving their original reasoning capability. This enables robot policies to effectively generalize from task reasoning to action execution.

3.1. Decoupled Latent Action Representation

Current latent action models predominantly use visual reconstruction as the training objective, which biases them toward learning image-space features rather than motion representations grounded in physical actions [14, 47, 57]. As a result, the learned latent actions remain far from executable robot actions, limiting the model's ability to rapidly adapt to new environments with few samples. Narrowing this gap is essential for establishing a reliable perception–action mapping and achieving efficient transfer to novel scenes. Moreover, existing methods [11, 57] typically entangle all visual variations, including both robot-induced motion and environment-induced changes within a single latent action representation. This entanglement introduces task-irrelevant signals (e.g., background motion or lighting fluctuations), weakens the correspondence between latent actions and true robot dynamics, and ultimately leads to inaccurate action predictions in manipulation tasks.

To address these issues, we propose a Universal Latent Action Learning framework that extracts latent actions from multi-frame observations under task-instruction guidance and jointly optimizes them via visual reconstruction and action generation objectives. This design enables the model to acquire physics-related priors (e.g., real-world distances and orientations) that more closely align with executable actions, thereby improving its transferability to downstream robotic manipulation tasks.

Specifically, we propose a **Decoupled Latent Action** Representation that separates the latent action Z_a into two components: the motion representation Z_{mot}, capturing the *active changes* driven by the robot's own motion, and the scene representation Z_{sce}, capturing the *passive scene changes* induced by environmental dynamics. This decomposition reduces task-irrelevant noise and establishes a clearer correspondence between robot motion, environmental variations, and latent action representations, thereby enhancing performance in downstream manipulation tasks. To extract two types of latent action presentations, we propose to leverage a pretrained vision-language model (VLM) due to its strong contextual understanding to reason about latent actions by integrating visual observations with language instructions. This process can be formulated as follows:

$$\{Z_{\text{sce}}, Z_{\text{mot}}\} = f_{\text{vlm}}(V_{t:t+k}, \ell), \tag{1}$$

where f_{vlm} denotes the VLM, which takes the visual frames as input from timestep t to $t+k$ along with the task instruction ℓ. In particular, we introduce two learnable latent action tokens, [CP_SCE] and [CP_MOT], into the VLM's vocabulary, allowing it to encode contextual information into structured scene representations Z_{sce} and motion representations Z_{mot}, respectively. To fully leverage the VLM's instruction-following ability for latent action summarization, we design an instruction-tuning template that guides the model in extracting the corresponding latent action representations from multi-frame sequences.

3.2. Unified Latent Action Decoding

To ensure that the latent actions focus on the dynamics changes from multiple frames, we further use them as conditional inputs to jointly guide the reconstruction of the future frame and the generation of inter-frame actions. In this way, the visual reconstruction constraint encourages the latent actions to capture observable scene variations, while the action generation objective provides physical-level guidance, enabling the model to establish a closer connection between the latent action and physical motions. Consequently, the model acquires universal latent actions that capture both visual dynamics and physical priors, enhancing prediction across diverse manipulation scenarios

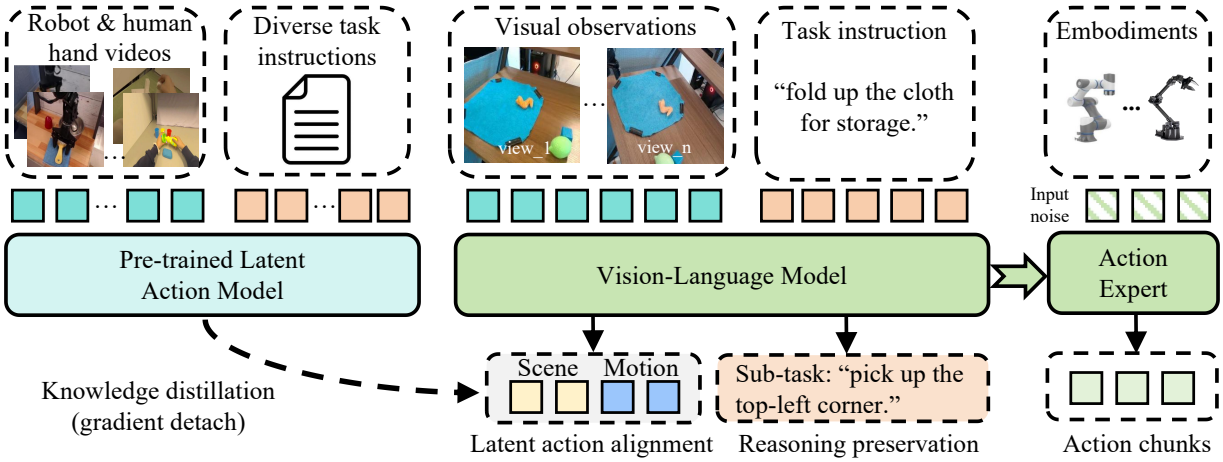

Figure 2. Illustration of the proposed latent action distillation approach for VLA models. By optimizing the VLMs with latent action alignment loss and reasoning preservation loss, we distill generalizable action representations learned from both robot and human hand demonstration videos, while simultaneously maintaining sub-task planning capabilities. This is followed by an action expert module for continuous action prediction.

and aligning more closely with real robot actions, thereby improving downstream task transferability.

Specifically, we propose a unified decoder, where scene and motion information are progressively fused through layer-wise bidirectional interactions, which can help the model learn latent actions that are better aligned with real robot dynamics. The decoder is initialized from pretrained image generation model SANA [55], which can leverage the pretrained model's powerful generative and contextual understanding capabilities. At each layer of the decoder, the scene and motion representations interact and exchange information, enabling progressive fusion between spatial and dynamic cues. Finally, the future visual frame V_{t+k} and inter-frame actions $A_{t:t+k}$ are decoded based on the fused scene and motion features. This bidirectional fusion mechanism allows scene dynamics to guide action generation, while motion tokens refine visual reconstruction, enabling mutual reinforcement between the two modalities.

3.3. Knowledge Distillation for VLA Models

Although the latent action model (LAM) effectively learns physically grounded latent action representations, its capabilities are limited to scene reconstruction and inter-frame action generation. To bridge this gap and transfer the learned knowledge to the vision–language–action (VLA) model, we propose a latent action knowledge distillation strategy. This approach enables the VLA to inherit motion understanding and physical priors from LAM while preserving its original vision–language reasoning abilities. As a result, the VLA can extract features closer to the action modality and acquire motion planning capabilities, facilitating efficient transfer to downstream manipulation tasks.

Specifically, given a pretrained latent action model (denoted as f_{lam}), a language instruction ℓ, and multiple frames $\{V_t\}_{t=1}^{T}$, the LAM first extracts latent action representations Z_a conditioned on the instruction:

$$Z_a = f_{\text{lam}}(\ell, \{V_t\}_{t=1}^{T}), \tag{2}$$

which captures the implicit correspondence between visual dynamics and task semantics. Meanwhile, the vision-language model (VLM) within the VLA model generates its own action representation conditioned only on the first frame V_1 and the same task instruction:

$$\hat{Z}_a = f_{\text{vlm}}(\ell, V_1), \tag{3}$$

where $\hat{Z}_a$ is expected to contain future motion information, which is generated by VLMs.

To align these two types of representations, we design a **Latent Action Alignment Loss** $\mathcal{L}_{\text{a}}$ that combines a reconstruction term and a distribution alignment term:

$$\mathcal{L}_{\text{a}} = \|\hat{Z}_a - Z_a\|_2^2 + \text{KL}\big(p(\hat{Z}_a) \,\|\, p(Z_a)\big), \tag{4}$$

where the first term enforces feature consistency and the second encourages distributional alignment, allowing the VLM to gain future frame forecasting capability. Unlike explicit action supervision in VLA models, latent actions are more embodiment-agnostic and naturally align with VLM representations. However, direct alignment may inadvertently compromise the VLM's inherent language understanding and reasoning abilities. To preserve these capabilities, we introduce a **Reasoning Preservation Loss** $\mathcal{L}_{\text{r}}$ to guide sub-task planning in robot manipulation tasks:

$$\mathcal{L}_{\text{r}} = -\sum_{i} \log p(w_{i+1} \mid w_{\le i}, \ell, V_1), \tag{5}$$

Table 1. Comparison of different VLA models across four tasks in two SIMPLER settings on the Google robot.

Google Robot	Method	Pick Coke Can	Move Near	Open/Close Drawer	Open Top Drawer and Place Apple	Avg
Visual Matching	RT-2-X [52]	78.7%	77.9%	25.0%	3.7%	46.3%
	OpenVLA [26]	18.0%	56.3%	63.0%	0.0%	34.3%
	π_0 [5]	87.3%	35.0%	72.6%	16.0%	52.7%
	SpatialVLA [43]	86.0%	77.9%	57.4%	0.0%	55.3%
	RoboVLM [32]	76.3%	79.0%	44.9%	27.8%	57.0%
	villa-X [11]	81.7%	55.4%	38.4%	-	-
	DD-VLA [30]	85.4%	67.5%	60.6%	-	-
	MemoryVLA [50]	90.7%	88.0%	84.7%	**47.2%**	77.2%
	Ours	**96.7%**	**91.7%**	**90.4%**	33.3%	**78.0%**
Variant Aggregation	RT-2-X [52]	82.3%	**79.2%**	35.3%	20.6%	54.4%
	OpenVLA [26]	60.8%	67.7%	28.3%	1.2%	39.3%
	π_0 [5]	85.2%	40.8%	42.1%	16.0%	46.0%
	SpatialVLA [43]	88.0%	72.7%	41.8%	6.3%	51.8%
	DD-VLA [30]	82.5%	64.6%	23.6%	-	-
	MemoryVLA [50]	80.5%	78.8%	53.2%	**58.3%**	67.7%
	Ours	**95.7%**	78.3%	**73.0%**	33.3%	**70.1%**

Table 2. Comparison of different VLA models across four tasks in the SIMPLER (Visual Matching) setting on the WidowX robot.

Method	Put Spoon on Towel	Put Carrot on Plate	Stack Green Block on Yellow Block	Put Eggplant in Yellow Basket	Avg
SpatialVLA [43]	16.7%	25.0%	29.2%	**100%**	42.7%
CogACT [27]	71.7%	50.8%	15.0%	67.5%	51.3%
π_0 [5]	62.5%	66.7%	25.0%	12.5%	41.7%
$\pi_{0.5}$ [20]	79.2%	58.3%	16.7%	66.7%	55.2%
villa-X [11]	48.3%	24.2%	19.2%	71.7%	40.8%
UniVLA [9]	52.8%	55.6%	2.8%	80.6%	47.9%
MemoryVLA [50]	75.0%	75.0%	37.5%	**100%**	71.9%
Ours	**95.8%**	**87.5%**	**83.3%**	83.3%	**87.5%**

which preserves the VLM's reasoning ability and enables it to autoregressively predict the $i+1$ token based on preceding tokens, generating coherent sub-task descriptions conditioned on the current frame and task instruction. Finally, the overall objective for latent action knowledge transfer is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{a}} + \lambda_{\text{r}} \cdot \mathcal{L}_{\text{r}}, \tag{6}$$

where λ_{r} balances the trade-off between latent action alignment and reasoning preservation and default to 0.5 in our experimental settings.

Action Expert Finetuning: after latent action knowledge distillation, the VLM not only retains its original vision–language reasoning capabilities, but also gains the motion planing abilities and generates features that are closely aligned with actions. However, these outputs are still latent representations and not directly executable as robot actions. Therefore, we further perform finetuning in both real-world and simulated robotic environments by incorporating an action expert, enabling precise generation of executable actions. To provide fine-grained supervision for action generation, we decompose the overall action loss into two components:$\mathcal{L}_{ee} + \mathcal{L}_{gripper}$, where $\mathcal{L}_{ee}$ denotes the loss for the end-effector's translation and rotation, computed using mean squared error. $\mathcal{L}_{gripper}$ denotes the loss for the gripper state, computed using binary cross-entropy to encourage more deterministic behavior.

4. Experiments

4.1. Implementation Details

Our latent action model (LAM) is pre-trained on a combined dataset of OXE [41], AgiBoT [8], and the human hand manipulation dataset EgoDex [18], encompassing a total of one million video episodes. For EgoDex, we lever-

Table 3. Comparison of different VLA models on the four LIBERO simulation environments.

Method	LIBERO-Goal	LIBERO-Object	LIBERO-Spatial	LIBERO-Long	Avg
Diffusion Policy [13]	68.3%	92.5%	78.3%	50.5%	72.4%
Octo [51]	84.6%	85.7%	78.9%	51.1%	75.1%
OpenVLA [26]	79.2%	88.4%	84.7%	53.7%	76.5%
TraceVLA [59]	75.1%	85.2%	84.6%	54.1%	74.8%
RDT [34]	68.2%	77.8%	60.2%	29.0%	58.8%
π_0 [5]	94.0%	97.8%	91.4%	85.4%	92.2%
UniVLA [9]	95.6%	96.8%	96.5%	92.0%	95.2%
villa-X [11]	91.5%	97.0%	97.5%	74.5%	90.1%
$\pi_{0.5}$ [20]	98.0%	98.2%	98.8%	92.4%	96.9%
MemoryVLA [50]	96.4%	98.4%	98.4%	93.4%	96.5%
Ours	**98.6%**	**98.8%**	**99.0%**	**95.4%**	**98.0%**

age its rich action annotations, including 3D positions and 6D orientations of both hands, as well as the 3D position of each fingertip, to provide fine-grained supervision for learning high-quality latent actions. Detailed specifications of the action space for both robot and human hand demonstrations are provided in the supplementary materials. The subsequent knowledge distillation stage is conducted on the same dataset. LAM pretraining and the distillation stage are performed on 16 NVIDIA A100 (40GB) GPUs for 14 and 7 days, respectively. The LAM encoder can be initialized from pretrained vision-language models such as PaliGemma [3] or InternVL3.5 [54]. The unified decoder is initialized from the pretrained image generation model SANA [55]. We train the model using Fully Sharded Data Parallel (FSDP) with a per-GPU batch size of 16 and a gradient accumulation step of 2, yielding an effective global batch size of 512. By default, LAM operates on 16-frame sequences and represents latent actions using 64 scene representations and 64 action representations. During knowledge distillation and fine-tuning, we use $\pi_{0.5}$ [20] as the default VLA backbone.

4.2. Manipulation Benchmark on SIMPLER

The SIMPLER [29] benchmark is designed to bridge the real-to-sim gap by recreating realistic scenarios for the Google Robot and WidowX Robot. We evaluate our method on the SIMPLER benchmark and compare it with a wide range of recent VLA models. Table 1 reports results on the Google Robot under the Visual Matching and Variant Aggregation settings. Across both evaluation protocols, our model consistently achieves the best overall performance. Under Visual Matching, our approach reaches 78.0% average success rate, outperforming all prior open-source models and improving over π_0 [5] by a significant margin (+25.3%). Notably, our method also surpasses the closed-source RT-2-X despite using fewer model parameters. Under Variant Aggregation, our model again sets a new state of the art with 70.1%, exceeding π_0 by 24.1% and RT-2-X by 15.7%. These results demonstrate the robustness of our model across different real-to-sim evaluation settings and its ability to generalize to visually altered scenes. As shown in Table 2, our method exhibits an even more pronounced advantage on the WidowX robot. It achieves an average success rate of 87.5%, substantially outperforming all existing VLA models. Compared with the strong baseline $\pi_{0.5}$ [20], our method achieves an improvement of 32.3%. More importantly, when compared with recent latent-action–based methods such as UniVLA [9] (47.9%) and villa-X [11] (40.8%), our model delivers gains of 39.6% and 46.7%, respectively. These results demonstrate that our method effectively transfers latent-action knowledge into the VLA domain, thereby improving the model's robustness and generalization across diverse tasks.

4.3. Manipulation Benchmark on LIBERO

The LIBERO [31] benchmark consists of four task suites, which are designed to study lifelong learning in robotic manipulation. We perform experiments on four task suites, each comprising 10 tasks with 50 human-teleoperated demonstrations. Specifically, *LIBERO-Spatial*, *LIBERO-Object* and *LIBERO-Goal* evaluate the understanding of the spatial relationships, object types and different task-oriented behaviors, respectively. *LIBERO-Long* test the ability to generalize the long-horizon tasks with different objects, layouts and goals. We fine-tune our model on the mixed LIBERO dataset for 60k steps with a batch size of 64. All methods are evaluated over 500 rollouts per task suite (i.e., 50 rollouts per task). As shown in Table 3, our method achieves the highest overall success rate of 98.0% across the four LIBERO environments. Compared with the baseline $\pi_{0.5}$, our method achieves a 3.0% improvement on *LIBERO-Long*, indicating that after latent action knowledge distillation, the VLA acquires stronger motion planning and future-state awareness, which substantially en-

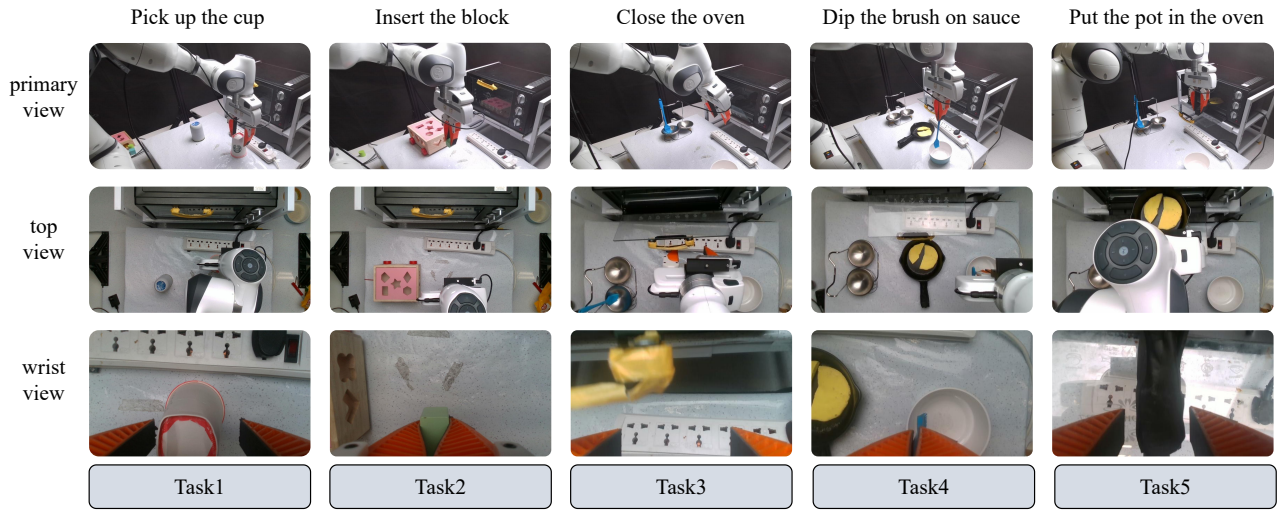

Figure 3. The real-robot Franka setup is equipped with multi-view observations. Tasks include pick-up, insertion, and so on, requiring both translational and rotational motions. In addition, we evaluate more practical scenarios involving interactions with real objects such as thin brushes, heavy frying pans, and real ovens.

Table 4. Success rate comparison across different tasks under varying the numbers of demonstrations for training.

Method	Task 1			Task 2			Task 3			Task 4			Task 5			Avg
	10	50	full	10	50	full	10	50	full	10	50	full	10	50	full	
π_0 [5]	0	0	10%	0	0	10%	40%	40%	60%	0	0	10%	0	0	20%	12.7%
$\pi_{0.5}$ [20]	0	0	20%	0	0	10%	60%	60%	80%	0	0	20%	20%	20%	20%	20.7%
Ours	**60%**	**80%**	**60%**	**20%**	**80%**	**40%**	**80%**	**100%**	**80%**	**20%**	**50%**	**80%**	**60%**	**60%**	**80%**	**63.3%**

hances its performance on long-horizon tasks.

4.4. Real-World Evaluation with Franka Robot

To rigorously evaluate the model's performance in real-world robot setups, which involve higher uncertainty and demand greater generalization from VLA models, we conduct a series of manipulation experiments using a Franka Research 3 robot with 7 degrees of freedom (DoFs) and a 1-DoF parallel gripper.

Task specifications: 1) To assess the instruction-following and visual understanding abilities, we first design a color discrimination task where the robot must correctly pick up a target cup when both a red and a blue cup are present in its view: *Pick up the cup* (**Task 1**). 2) To further examine the fine-grained manipulation and physical reasoning capabilities, we introduce four more challenging tasks: *Put the building block into the corresponding slot* (**Task 2**), *Close the oven* (**Task 3**), *Dip the brush in the sauce* (**Task 4**), and *Put the pot in the oven* (**Task 5**). **Task 2** and **Task4** require multi-stage control and precise spatial understanding, while **Task 3** and **Task 5** demand delicate gripper control. For example, closing the oven door requires accurately grasping the handle and applying force along the correct motion direction.

Note that each task comprises 100 demonstrations, collected via human expert teleoperation. To evaluate the model's few-shot transfer capability, we train it using subsets of 10, 50, and all available demonstrations for each task. We compare our approach against π_0 and $\pi_{0.5}$. Table 4 summarizes the success rates of the models across five real-world manipulation tasks under varying demonstration sizes. Our method consistently outperforms the baselines across nearly all tasks and training settings. For the same number of the training samples (e.g., 50-shot), all models are trained using the same batch size and number of GPUs, with the same amount of training steps and each task is evaluated over 10 trials.

For the color discrimination task (**Task 1**), our model achieves a 60% success rate with only 10 demonstrations and 80% with 50-shot, whereas both baselines fail entirely in the few-shot settings and reach at most 20% with the full dataset. Interestingly, the 50-shot performance surpasses that of fine-tuning with all available data. This likely results from the full dataset containing redundant action patterns,

Table 5. Impact of each component, evaluated in the SIMPLER benchmark. **UAD**, **DLA** denotes the unified action decoder, the decoupled latent action representations, respectively.

	UAD	DLA	Put Spoon on Towel	Put Carrot on Plate	Stack Green Block on Yellow Block	Put Eggplant in Yellow Basket	Avg
UniVLA-style			41.6%	54.2%	45.8%	62.5%	51.0%
Ours-v1		✓	70.8%	66.7%	37.5%	62.5%	59.4%
Ours-v2	✓		70.8%	66.7%	41.7%	66.7%	61.5%
Ours-v3	✓	✓	95.8%	87.5%	83.3%	83.3%	87.5%

which cause the latent actions to encode irrelevant variations. Consequently, the VLA model may produce slightly inaccurate actions, reducing overall success. In contrast, the 50-shot subset is more concise and cleaner, allowing latent actions to focus on the core, task-relevant motion features.

For the multi-stage control task (**Task 2**), where the robot must first pick up a building block and then place it into the corresponding slot, our model demonstrates strong few-shot learning capability, achieving 80% success with only 50 demonstrations, while the baselines show no success until trained on the full dataset. This task is particularly challenging as it requires both sequential reasoning and precise spatial alignment—the robot must grasp the block accurately and position it correctly in the slot. The strong few-shot performance indicates that our latent actions capture essential task dynamics and structure, enabling the VLA model to plan and execute multi-step manipulations effectively with limited data. Similarly, in the fine-grained manipulation tasks (**Task 3–Task 5**), which demand precise gripper control and spatial reasoning, our method consistently outperforms the baselines. For example, **Task 4** challenges the model with small-object manipulation and fine-grained end-effector control, as the robot must stably grasp a thin handle and execute a targeted dipping motion without disturbing the bowl. In this task, our method achieves a 50% success rate with only 50 demonstrations, while $\pi_{0.5}$ fails to complete the task. When using the full dataset, our approach surpasses $\pi_{0.5}$ by 60%. **Task 5** requires precisely grasping the center of the pan handle, otherwise the task will fail. Our method outperforms $\pi_{0.5}$ in both the few-shot and full-shot settings. This demonstrates that latent action learning effectively equips the model with structured motion priors and fine-grained physical understanding, enabling stable small-object manipulation even under few-shot settings.

4.5. Components Analysis

The ablation results in Table 5 show that both the decoupled latent action representations (DLA) and the unified action decoder (UAD) play essential and complementary roles in improving manipulation performance. Starting from the UniVLA-style baseline of 51.0%, introducing DLA alone yields a clear gain to 59.4% by isolating motion-critical cues from irrelevant environment changes, allowing the model to form cleaner and structured latent actions. Adding only UAD further improves the average to 61.5%, as the decoder strengthens the mapping between latent actions and executable robot actions, reducing modality gap during action generation. When both components are combined, the model achieves a substantial jump to 87.5%, consistently outperforming all other variants across every task. This strong synergy arises because DLA provides structured, manipulation-relevant latent actions, while UAD injects physical priors into the latent action learning process, enabling the robot to generate more accurate and physically consistent action predictions.

5. Conclusion

In summary, we propose a universal latent action learning framework, **LatBot**, and demonstrates that learning transferable latent actions from large-scale object manipulation videos (e.g., robot and human hand), substantially enhances generalization in downstream robotic tasks. By integrating task instructions with multi-frame observations, jointly optimizing future frame reconstruction and action sequence prediction, and disentangling latent actions into motion and scene tokens, our framework effectively captures rich physical priors while filtering out irrelevant dynamics. Experiments show that distilling these latent actions into VLA models yields strong performance across both simulated and real-world robotic platforms. Notably, even with only a few real-world demonstrations on a Franka robot, our method shows that latent actions offer a robust, generalizable representation for complex manipulation tasks, including pick-and-place of thin objects (e.g., brushes) and precise block insertion requiring fine-grained motions.

These results highlight a key insight: explicitly incorporating physical priors and disentangling motion from environmental changes significantly enhances the transferability of learned latent action representations. For future work, we aim to extract additional latent tokens from larger and more diverse manipulation video datasets, further scaling VLA models and exploring their potential for more complex, long-horizon, and multi-embodiment robotic tasks.

Acknowledgement

This work was supported in part by Science and Technology Innovation (STI) 2030—Major Projects under Grant 2022ZD0208700, and National Natural Science Foundation of China under Grant 62376264.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[2] Suneel Belkhale, Yuchen Cui, and Dorsa Sadigh. Hydra: Hybrid robot actions for imitation learning. In *Conference on Robot Learning*, pages 2113–2133. PMLR, 2023.

[3] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

[4] Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *arXiv preprint arXiv:2310.10639*, 2023.

[5] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[6] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[7] Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

[8] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

[9] Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

[10] Lawrence Yunliang Chen, Simeon Adebola, and Ken Goldberg. Berkeley UR5 demonstration dataset. https://sites.google.com/view/berkeley-ur5/home.

[11] Xiaoyu Chen, Hangxing Wei, Pushi Zhang, Chuheng Zhang, Kaixin Wang, Yanjiang Guo, Rushuai Yang, Yucen Wang, Xinquan Xiao, Li Zhao, et al. Villa-x: enhancing latent action modeling in vision-language-action models. *arXiv preprint arXiv:2507.23682*, 2025.

[12] Yi Chen, Yuying Ge, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent motion token as the bridging language for robot manipulation. *arXiv preprint arXiv:2412.04445*, 8, 2024.

[13] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[14] Zichen Cui, Hengkai Pan, Aadhithya Iyer, Siddhant Haldar, and Lerrel Pinto. Dynamo: In-domain dynamics pretraining for visuo-motor control. *Advances in Neural Information Processing Systems*, 37:33933–33961, 2024.

[15] Zichen Jeff Cui, Yibin Wang, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. From play to policy: Conditional behavior generation from uncurated robot data. *arXiv preprint arXiv:2210.10047*, 2022.

[16] Shivin Dass, Jullian Yapeter, Jesse Zhang, Jiahui Zhang, Karl Pertsch, Stefanos Nikolaidis, and Joseph J. Lim. Clvr jaco play dataset, 2023.

[17] Minho Heo, Youngwoon Lee, Doohyun Lee, and Joseph J Lim. Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation. *The International Journal of Robotics Research*, 44(10-11):1863–1891, 2025.

[18] Ryan Hoque, Peide Huang, David J Yoon, Mouli Sivapurapu, and Jian Zhang. Egodex: Learning dexterous manipulation from large-scale egocentric video. *arXiv preprint arXiv:2505.11709*, 2025.

[19] Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. *arXiv preprint arXiv:2412.14803*, 2024.

[20] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[21] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.

[22] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on robot learning*, pages 651–673. PMLR, 2018.

[23] Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. Your large vision-language model only needs a few attention heads for visual grounding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9339–9350, 2025.

[24] Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned

language models. In *Forty-first International Conference on Machine Learning*, 2024.

[25] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[26] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[27] Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024.

[28] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. *arXiv preprint arXiv:2311.01378*, 2023.

[29] Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024.

[30] Zhixuan Liang, Yizhuo Li, Tianshuo Yang, Chengyue Wu, Sitong Mao, Liuao Pei, Xiaokang Yang, Jiangmiao Pang, Yao Mu, and Ping Luo. Discrete diffusion vla: Bringing discrete diffusion to action decoding in vision-language-action policies. *arXiv preprint arXiv:2508.20072*, 2025.

[31] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

[32] Huaping Liu, Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, and Hanbo Zhang. Towards generalist robot policies: What matters in building vision-language-action models. 2025.

[33] Huihan Liu, Soroush Nasiriany, Lance Zhang, Zhiyao Bao, and Yuke Zhu. Robot learning on the job: Human-in-the-loop autonomy and learning during deployment. *The International Journal of Robotics Research*, 44(10-11):1727–1742, 2025.

[34] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

[35] Jianlan Luo, Charles Xu, Xinyang Geng, Gilbert Feng, Kuan Fang, Liam Tan, Stefan Schaal, and Sergey Levine. Multi-stage cable routing through hierarchical imitation learning. *IEEE Transactions on Robotics*, 40:1476–1491, 2024.

[36] Jianlan Luo, Charles Xu, Fangchen Liu, Liam Tan, Zipeng Lin, Jeffrey Wu, Pieter Abbeel, and Sergey Levine. Fmb: a functional manipulation benchmark for generalizable robotic learning. *The International Journal of Robotics Research*, 44 (4):592–606, 2025.

[37] Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*, 2023.

[38] Ajay Mandlekar, Jonathan Booher, Max Spero, Albert Tung, Anchit Gupta, Yuke Zhu, Animesh Garg, Silvio Savarese, and Li Fei-Fei. Scaling robot supervision to hundreds of hours with roboturk: Robotic manipulation dataset through human reasoning and dexterity. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1048–1055. IEEE, 2019.

[39] Oier Mees, Jessica Borja-Diaz, and Wolfram Burgard. Grounding language with visual affordances over unstructured data. *arXiv preprint arXiv:2210.01911*, 2022.

[40] Soroush Nasiriany, Tian Gao, Ajay Mandlekar, and Yuke Zhu. Learning and retrieval from prior data for skill-based imitation learning. *arXiv preprint arXiv:2210.11435*, 2022.

[41] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya 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)*, pages 6892–6903. IEEE, 2024.

[42] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

[43] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.

[44] Gabriel Quere, Annette Hagengruber, Maged Iskandar, Samuel Bustamante, Daniel Leidner, Freek Stulp, and Jörn Vogel. Shared control templates for assistive robotics. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 1956–1962. IEEE, 2020.

[45] Juntao Ren, Priya Sundaresan, Dorsa Sadigh, Sanjiban Choudhury, and Jeannette Bohg. Motion tracks: A unified representation for human-robot transfer in few-shot imitation learning. *arXiv preprint arXiv:2501.06994*, 2025.

[46] Saumya Saxena, Mohit Sharma, and Oliver Kroemer. Multi-resolution sensing for real-time control with vision-language models. In *2nd Workshop on Language and Robot Learning: Language as Grounding*, 2023.

[47] Dominik Schmidt and Minqi Jiang. Learning to act without actions. *arXiv preprint arXiv:2312.10812*, 2023.

[48] Nur Muhammad Mahi Shafiullah, Anant Rai, Haritheja Etukuru, Yiqian Liu, Ishan Misra, Soumith Chintala, and Lerrel Pinto. On bringing robots home. *arXiv preprint arXiv:2311.16098*, 2023.

[49] Rutav Shah, Roberto Martín-Martín, and Yuke Zhu. Mutex: Learning unified policies from multimodal task specifications. *arXiv preprint arXiv:2309.14320*, 2023.

[50] Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. Memoryvla: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2508.19236*, 2025.

[51] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[52] Quan Vuong, Sergey Levine, Homer Rich Walke, Karl Pertsch, Anikait Singh, Ria Doshi, Charles Xu, Jianlan Luo, Liam Tan, Dhruv Shah, et al. Open x-embodiment: Robotic learning datasets and rt-x models. In *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition@ CoRL2023*, 2023.

[53] Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pages 1723–1736. PMLR, 2023.

[54] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

[55] Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, et al. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*, 2024.

[56] Ge Yan, Kris Wu, and Xiaolong Wang. ucsd kitchens dataset. 2023.

[57] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. *arXiv preprint arXiv:2410.11758*, 2024.

[58] Chuheng Zhang, Tim Pearce, Pushi Zhang, Kaixin Wang, Xiaoyu Chen, Wei Shen, Li Zhao, and Jiang Bian. What do latent action models actually learn? *arXiv preprint arXiv:2506.15691*, 2025.

[59] Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. *arXiv preprint arXiv:2412.10345*, 2024.

[60] Gaoyue Zhou, Victoria Dean, Mohan Kumar Srirama, Aravind Rajeswaran, Jyothish Pari, Kyle Hatch, Aryan Jain, Tianhe Yu, Pieter Abbeel, Lerrel Pinto, et al. Train offline, test online: A real robot learning benchmark. *arXiv preprint arXiv:2306.00942*, 2023.

[61] Zhongyi Zhou, Yichen Zhu, Junjie Wen, Chaomin Shen, and Yi Xu. Vision-language-action model with open-world embodied reasoning from pretrained knowledge. *arXiv preprint arXiv:2505.21906*, 2025.

[62] Xinghao Zhu, Ran Tian, Chenfeng Xu, Mingxiao Huo, Wei Zhan, Masayoshi Tomizuka, and Mingyu Ding. Fanuc manipulation: A dataset for learning-based manipulation with fanuc mate 200id robot, 2023.

[63] Yifeng Zhu, Peter Stone, and Yuke Zhu. Bottom-up skill discovery from unsegmented demonstrations for long-horizon robot manipulation. *IEEE Robotics and Automation Letters*, 7(2):4126–4133, 2022.

[64] Yifeng Zhu, Abhishek Joshi, Peter Stone, and Yuke Zhu. Viola: Imitation learning for vision-based manipulation with object proposal priors. In *Conference on Robot Learning*, pages 1199–1210. PMLR, 2023.

[65] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.

LatBot: Distilling Universal Latent Actions for Vision-Language-Action Models

Supplementary Material

6. Implementation Details

6.1. Dataset Details

We pre-train the latent action model using a combination of robot and human manipulation data, with a total of 1 million video episodes. For robotic manipulation data, we use the OXE [41], DROID [25], and AgiBoT [8] datasets. For human hand manipulation, we utilize EgoDex [18], which provides detailed hand pose annotations and represents the largest and most diverse dataset for dexterous human manipulation to date. EgoDex provides full bimanual skeletal joints, where each action at a time step is represented by the 3D position of each wrist, the 6D wrist orientation, and the 3D positions of the five fingertips on each hand, resulting in a 48-dimensional action vector. To unify the action representations of robots and human hands during latent action pretraining, we design a **Unified Action Space** with a total dimensionality of 44:

- **Dimensions 1–7 (Left hand / left arm):** Changes in x, y, z, Euler orientation, and gripper state.
- **Dimensions 8–14 (Right hand / right arm):** The same set of changes for the right hand or right robotic arm.
- **Dimensions 15–44 (Bimanual fingertips):** 3D position changes of the ten fingertips (five for each hand), totaling 30 dimensions.

We additionally define a **Unified State Space** for both robots and human hands:

- **Dimensions 1–8 (Left hand / left arm):** Current-time x, y, z, quaternion orientation (4D), and gripper state.
- **Dimensions 9–16 (Right hand / right arm):** The same information for the right hand or right robotic arm.
- **Dimensions 17–46 (Bimanual fingertips states):** 3D positions of ten fingertips from both hands at the current time step.

Since EgoDex provides 6D orientations, we convert them to Euler angles for the unified action space and to quaternions for the unified state space. The detailed composition of the datasets and mixture weights are listed in Table. 6.

Table 6. Mixture of datasets used during pretraining, including OXE [41], DROID [25], and EgoDex [18].

Dataset Name	Ratio
Fractal [6]	12.8%
Kuka [22]	12.8%
Bridge [53]	11.8%
Taco Play [39]	2.7%
Jaco Play [16]	0.4%
Berkeley Cable Routing [35]	0.2%
Roboturk [38]	2.1%
Viola [64]	0.8%
Berkeley Autolab UR5 [10]	1.1%
Toto [60]	1.8%
Language Table [37]	4.4%
Stanford Hydra Dataset [2]	4.6%
Austin Buds Dataset [63]	0.2%
NYU Franka Play Dataset [15]	0.6%
Furniture Bench Dataset [17]	2.5%
UCSD Kitchen Dataset [56]	$< 0.1\%$
Austin Sailor Dataset [40]	2.2%
Austin Sirius Dataset [33]	1.7%
DLR EDAN Shared Control [44]	$< 0.1\%$
IAMLab CMU Pickup Insert [46]	0.9%
UTAustin Mutex [49]	2.2%
Berkeley Fanuc Manipulation [62]	7.8%
CMU Stretch [40]	1.5%
BC-Z [21]	6.8%
FMB Dataset [36]	7.1%
DobbE [48]	1.4%
DROID [25]	$< 0.1\%$
AgiBoT-α [8]	6.3%
EgoDex [18]	11.1%

6.2. Model Details

Our latent action model employs a vision-language model (VLM) as the latent action encoder, which summarizes the inter-frame dynamics into a sequence of latent action representations under language guidance. This VLM component can be any pretrained vision-language model, such as PaliGemma [3] or InternVL [54]. In this work, we default to using InternVL3.5-2B. For the latent action decoder, we adopt an architecture similar to SANA-1.6B [55] and initialize it with the corresponding pretrained weights. The instruction-tuning template used to summarize the latent action representations, including both the question and answer components, is defined as follows:

> *Given the instruction "{sent}" and the video frames, reason about what happens within this time span. Explain how the overall scene changes, and identify the temporal dependencies between consecutive frames. Highlight the actions, interactions, and transitions that drive the scene's evolution. Answer: Scene evolution description: [CP_SCE]. Action dynamics description: [CP_MOT].*

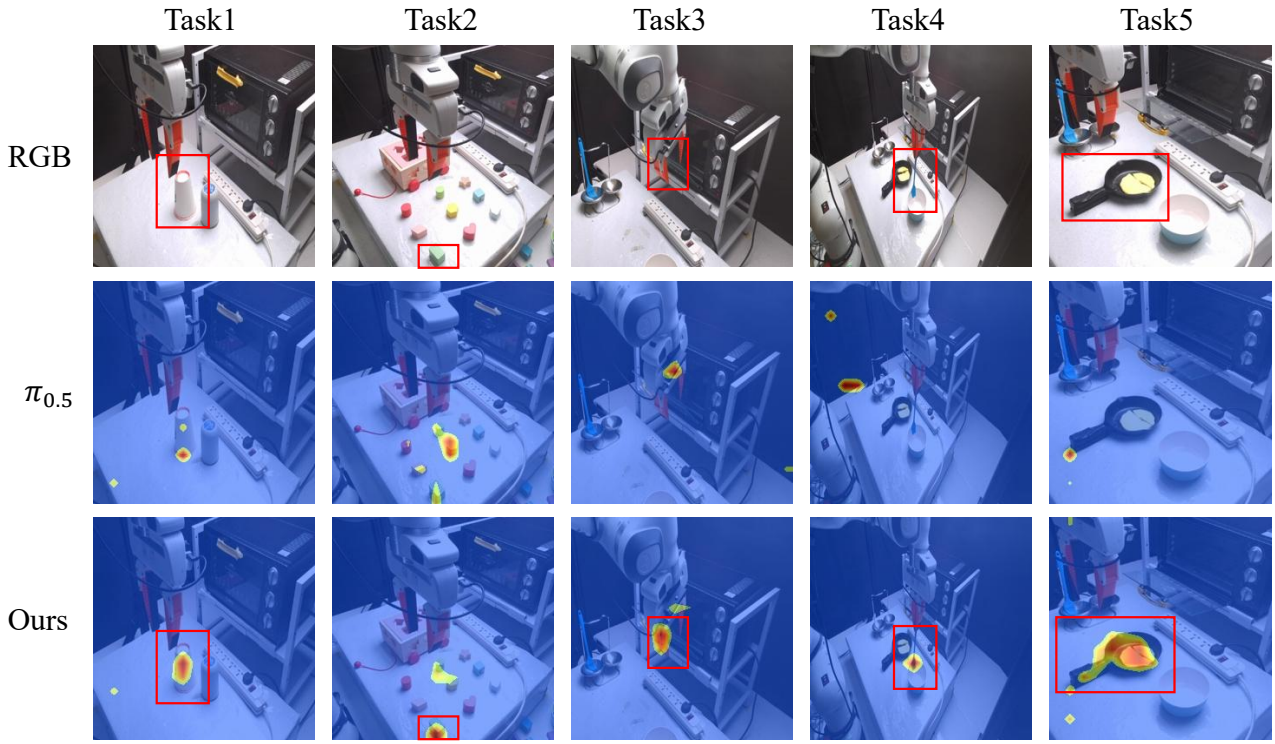

Figure 4. Analysis of the effects the latent-action distillation on the VLM. Following [23], we visualize the attention maps between the final text token and the visual features for both $\pi_{0.5}$ and our model across different real-robot tasks. The red bounding boxes in the RGB images mark the task-specific target, while the red boxes on our model's attention maps highlight the regions with the strongest activations. The results show that after latent-action distillation, the VLM of our model exhibits enhanced spatial grounding capabilities, with its attention maps consistently concentrated within the red box.

During the knowledge distillation stage, we use the VLM from $\pi_{0.5}$ as our student model and the teacher model is the pretrained latent action model.

6.3. Training Details

During the latent action pretraining stage, we first train the model on a mixed dataset containing both robot and human manipulation data. The pretraining runs for 14 days on 16 RTX A100 GPUs (40GB), with a per-GPU batch size of 16 and a gradient accumulation step of 2, resulting in a total batch size of 512. In this stage, all parameters of the latent action model are optimized except for the vision encoder of the VLM, which remains frozen. We use the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.95$. The learning rate is initialized at 1.0×10^{-4}, followed by a 2,000-step warm-up phase and a cosine decay schedule that gradually anneals it to a minimum of 2.5×10^{-6}.

For the latent action distillation stage, we use the same dataset as in pretraining. We jointly fine-tune all parameters of the student model (VLM from the vision-language-action model) while keeping the teacher model (VLM from the latent action model) frozen. The learning rate schedule follows the same configuration as in pretraining. The distillation process lasts 7 days on 16 RTX A100 GPUs (40GB), with a per-GPU batch size of 8 and a gradient accumulation step of 2, yielding a total batch size of 256. During both simulation and real-robot fine-tuning, we jointly fine-tune the vision encoder, the VLM, and the action expert components of the VLA. Following [20], we apply quantile normalization for action and state normalization.

7. Additional Analysis

The effect of the latent action on the VLM. In real-robot experiments, we observe that our model exhibits strong spatial understanding, enabling it to accurately place blocks into their corresponding slots. To further examine the effect of latent action knowledge on the VLM, we analyze the visual grounding capability of the model before and after distillation. Since LLMs decode in an auto-regressive manner, information gradually accumulates from earlier tokens to later ones, causing the final text token to incorporate the semantic context of the entire instruction [23]. Therefore, the query vector of the last input text token serves as a representative probe for evaluating sentence-level grounding.

We use the query vector of the last input text token as a sentence-level representation for computing attention over image features.

Specifically, given the query token, we extract the attention weights from the query to all image tokens across all layers and heads. For each attention head, we take the first P^2 entries and reshape them into a spatial attention map with size of $P \times P$, where P denotes the patch size. The attention map is then binarized by assigning value 1 to elements above the mean and 0 otherwise. Next, we detect connected components $\{C_i\}_{i=1}^{N}$ using 8-neighborhood connectivity and compute the spatial entropy:

$$H = -\sum_{i=1}^{N} P(C_i) \log P(C_i), \tag{7}$$

where $P(C_i) = \frac{|C_i|}{\sum_{j=1}^{N} |C_j|}$. An attention map is considered more spatially localized when it exhibits lower spatial entropy. For visualization, we report the attention map with the lowest spatial entropy among all layers and heads, as it best captures the model's most concentrated grounding behavior.

As shown in Fig. 4, we compare the attention maps of the last text token over image features between $\pi_{0.5}$ and our method across various real-world robotic tasks. Results show that after latent-action distillation, the VLM can localize task-relevant targets more accurately based on the instruction. When distractors are present (Task2), it exhibits an even stronger response to the true target (reflected by darker attention regions).