DexVLA: Vision-Language Model with Plug-In Diffusion Expert for General Robot Control

Junjie Wen[*,1,2] **Yichen Zhu**[*,1,†] **Jinming Li**[1,3] **Zhibin Tang**[1] **Chaomin Shen**[2] **Feifei Feng**[1]

https://dex-vla.github.io/

Figure 1: **Dexterous skills in diverse tasks and scenarios.** Our proposed DexVLA method enables generalized dexterous manipulation across multiple embodiments in diverse scenarios.

Abstract: Enabling robots to perform diverse tasks across varied environments is a central challenge in robot learning. While vision-language-action (VLA) models have shown promise for generalizable robot skills, realizing their full potential requires addressing limitations in action representation and efficient training. Current VLA models often focus on scaling the vision-language model (VLM) component, while the action space representation remains a critical bottleneck. This paper introduces DexVLA, a novel framework designed to enhance the efficiency and generalization capabilities of VLAs for complex, long-horizon tasks across diverse robot embodiments. DexVLA features a novel diffusion-based action expert, scaled to one billion parameters, designed for cross-embodiment learning. A novel embodiment curriculum learning strategy facilitates efficient training: (1) pre-training the diffusion expert on cross-embodiment data, (2) aligning the VLA model to specific embodiments, and (3) post-training for rapid adaptation to new tasks. We conduct comprehensive experiments across multiple embodiments, including single-arm, bimanual, and dexterous hand, demonstrating DexVLA's

[*]: denotes equal contribution. [†] denotes corresponding author.
[1]Midea Group, [2]East China Normal University, [3]Shanghai University

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

adaptability to challenging tasks without task-specific adaptation, its ability to learn dexterous skills on novel embodiments with limited data, and its capacity to complete complex, long-horizon tasks using only direct language prompting, such as laundry folding. In all settings, our method demonstrates superior performance compared to state-of-the-art models like OpenVLA and π_0.

Keywords: Vision-Language-Action Model, Robotic Manipulation

1 Introduction

Enabling robots to perform diverse tasks across varied environments is a central challenge in robotics. Achieving versatility—the ability to solve a variety of tasks across diverse environments, while adapting to language commands, environmental constraints, and unexpected disruptions—is even more demanding. Imitation learning [1, 2, 3, 4, 5, 6, 7], particularly through vision-language-action (VLA) models [8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19], has shown promise in enabling generalizable skills.

However, realizing the vision of omnipotent robot foundation models faces persistent challenges. Two key bottlenecks hinder progress: 1) Data scarcity: State-of-the-art models, like OpenVLA [20] and Octo [21], rely on massive datasets like the Open-X Embodiment dataset (4,000 hours) [22] or even larger corpora like the 10,000-hour dataset used by π_0 & $\pi_{0.5}$ [8, 13]. Collecting such data through human demonstrations is extremely costly and labor-intensive [23, 24, 25, 26]. (2) Architectural imbalance: current VLA models often prioritize scaling the vision-language model (VLM) component, i.e., OpenVLA uses 7B VLM and π_0 uses 3B VLM. Despite its enhanced visual and linguistic understanding through internet-scale data pretraining, the VLM component remains disconnected from the embodied, sensorimotor context of robotic action.

This paper introduces Plug-in **D**iffusion **Ex**pert for **V**ision-**L**anguage-**A**ction models, namely **DexVLA**, a novel framework designed to enhance the data efficiency and generalization capabilities of VLA for complex, long-horizon tasks across diverse robot embodiments. We achieve this through two key innovations:

1) Billion-Parameter Diffusion Expert: Recognizing the limitations of conventional action experts, particularly in handling cross-embodiment data, we propose a new diffusion-based action expert. The diffusion expert utilizes a multi-head architecture, with each head corresponding to a specific embodiment, enabling effective learning across diverse morphologies. Furthermore, we scale the model size of the diffusion expert to one billion parameters, a substantial increase from the conventional multi-million parameter scale. This scaling significantly enhances the model's capacity to learn intricate motor skills and control policies from diverse and extensive data.

2) Embodied Curriculum Learning: A three-stage training strategy that progressively learns harder tasks. This is conceptually similar to how human learn, which starts with simple tasks, and then gradually introduces complexity to avoid overwhelming the learner.

Stage 1: The *cross-embodiment pre-training* stage focuses on learning low-level, embodiment-agnostic motor skills. In this stage, we pre-train only the diffusion expert using cross-embodiment data, without involving the vision-language models.

Stage 2: The *embodiment-specific alignment* is a stage that analogy to "adapt to your body". Specifically, it bridges abstract vision-language representations to the physical constraints of a specific robot. Remarkably, this stage alone enables the model to complete a variety of tasks, such as shirt folding and bin picking on in-domain objects.

Stage 3: The *task-specific adaptation* aims to make the robot master complex tasks. These tasks include completing long-horizon tasks and generalizing to novel objects.

While the model has learned from diverse robot behaviors and progressively developed dexterous skills for complex tasks, it faces limitations in very long-horizon, contact-rich scenarios such as folding crumpled shirts or executing continuous bin-picking. Prior approaches often rely on high-level policy models; for example, π_0 use SayCan to update instructions every two seconds. In

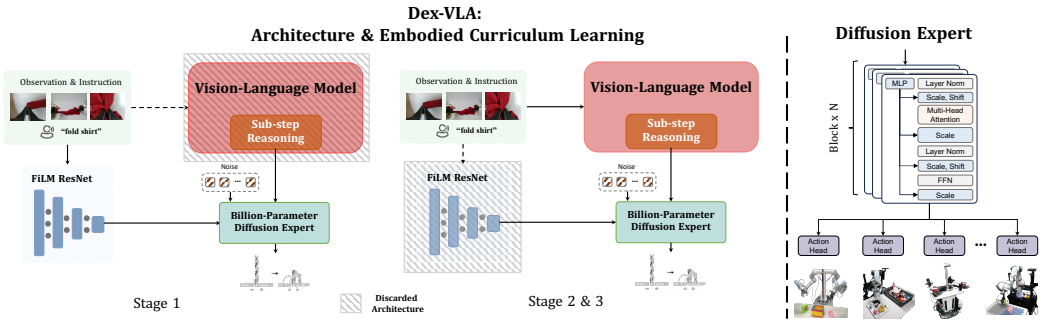

Figure 2: **DexVLA architecture and embodied curriculum learning.** Our model employs a three-stage training process. **Stage 1 (left)** trains the Diffusion Expert independently, without the VLM. **Stages 2 and 3 (middle)** integrate the Diffusion Expert with a VLM, discarding the visual and language components within the expert. **The Diffusion Expert (right)** uses multiple heads for cross-embodiment learning.

contrast, we propose leveraging the innate reasoning abilities of the VLA model to directly guide robot action. We train the model using demonstrations annotated with *substep reasoning* — for instance, breaking "fold the shirt" into "smooth wrinkles," "align sleeves," and "secure folds" — enabling it to learn disentangled action representations that map language sub-instructions to precise motor primitives.

We evaluate DexVLA across diverse embodiments, including single-arm, bimanual, dexterous hand, and mobile bimanual robots, demonstrating its effectiveness on a variety of tasks. DexVLA achieves high success rates on many tasks without task-specific adaptation. For example, it achieves a near full score in folding flattened shirts. It can also learn dexterous skills on novel embodiments with fewer than 100 demonstrations, such as pouring drinks with a dexterous hand and packing on a bimanual robot. Furthermore, when directly prompting VLA model on completing complex, long-horizon tasks like laundry folding, DexVLA outperforms π_0 by a large margin. Importantly, our model is pre-trained on only 100 hours of demonstration data and runs at 60Hz on a single Nvidia A6000 GPU, enabling cost-efficient training and fast inference.

2 Method

2.1 Model Architecture

Our DexVLA model is primarily based on a transformer language model backbone. We use Qwen2-VL [27] as the base VLM model. Following the common framework of VLM models, we employ image encoders to project the robot's image observations into the same embedding space as the language tokens. For multiple camera views, these visual tokens are concatenated. The VLM component generates two outputs: reasoning tokens and action tokens. The action tokens are passed through a projection module, consisting of two linear layers with LayerNorm. This module is analogous to the connectors designed in vision-language models like LLaVA [28], and serves to transform the VLM's embedding space to align with the input requirements of the action expert. The reasoning tokens are injected into the policy model using FiLM layers, which scale and shift the parameters of the projection layers within the policy. Consequently, the model can autonomously generate reasoning and leverage this reasoning within the diffusion expert to guide action generation. The overview is presented in Figure 2.

Building diffusion expert. Since action experts dominate the learning process of robot's action, it is essential to design a good neural architecture for better visuomotor policy learning. We utilized the Scale Diffusion Policy (ScaleDP [29]), a variant of the Diffusion Policy in Transformer architecture, where the largest version of ScaleDP is up to 1B parameters. However, the naive ScaleDP is not designed for cross-embodiment pre-training. We put a multi-head output to enable pre-training on ScaleDP with various robot configurations. Each head is responsible for a single robot configuration. This setup is similar to Octo [21].

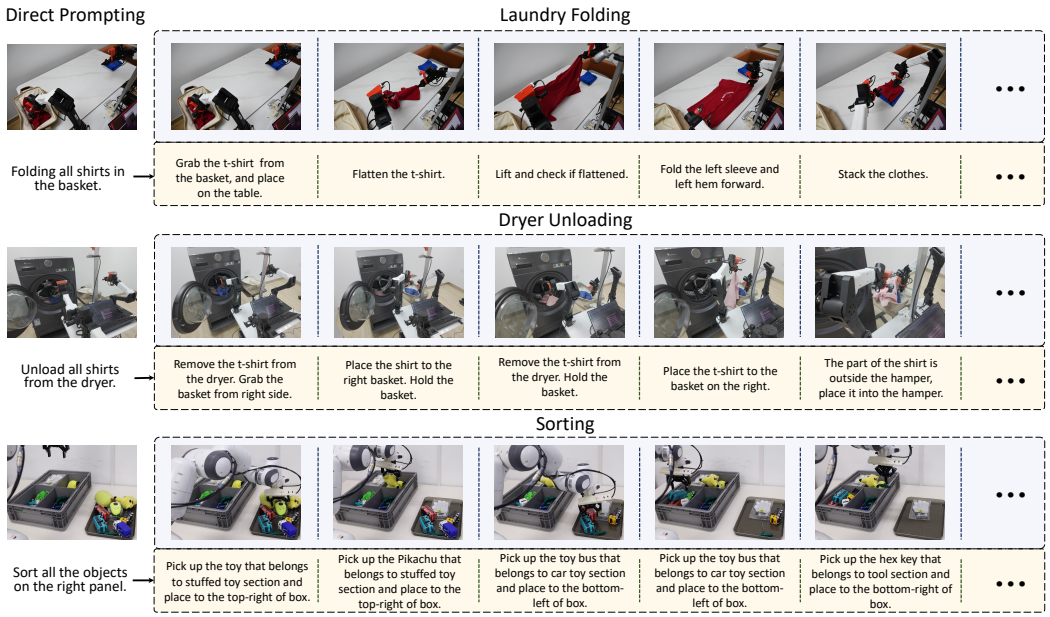

Figure 3: **Example of direct prompting for long-horizon tasks.** The figure shows three tasks, **laundry folding (top)**, **dryer unloading (middle)**, **sorting (bottom)**. Our DexVLA breaks down raw instructions into sub-steps automatically. Success in these tasks necessitates not only dexterity but also the capacity to decompose direct prompts into implicit multi-step reasoning and to comprehend the visual context.

Figure 4: Our experiment includes various robot types: bimanual UR5e, Franka, bimanual AgileX, and Franka with dexterous hands.

Table 1: **Detailed hyperparameters for training DexVLA.**

Hyperparameters	Stage 1	Stage 2	Stage 3
Learning rate	1e-4	2e-5	2e-5
LR scheduler	Constant	Constant	Cosine
Weight decay	0.0	0.0	0.0
Optimizer	AdamW($\beta_1 = 0.9$,β_2=0.95)		
Training epochs	5	5	5
Data	Cross-embodied	Embodied-specific	Task-specific

Training objectives. Given a batch of input sequences, the overall training loss is defined as a weighted combination of the diffusion loss (L_{diff}) and the next-token prediction loss (L_{ntp}), which are $L = L_{diff} + \alpha L_{ntp}$. For all experiments, we set $\alpha = 1$, as we observe that L_{ntp} converges during the early stages of training. Consequently, this setup allows the model to primarily focus on learning robot action prediction based on reasoning and instructions.

2.2 Embodied Curriculum Learning

Curriculum learning is an training strategy where a system learns tasks in a progression from simple to complex, mirroring how humans acquire skills. Our three-stage training strategy implements an embodied curriculum, where the policy network first learns generalizable motor skills from cross-embodiment data (Stage 1), then adapts to its specific physical form (Stage 2), and finally refines task-specific behaviors (Stage 3). This mirrors human skill acquisition, where foundational abilities (e.g., grasping) precede specialized expertise (e.g., folding clothes).

A well-designed training strategy is critical for optimizing deep neural networks. Approaches that align with a network's inherent training dynamics ensure more efficient and effective data utilization. The DexVLAtargets general robotic control by integrating a VLM with a diffusion expert. Leveraging its modular architecture—which combines two distinct components—we propose a three-stage training strategy that systematically addresses: (1) learning dexterous manipulation skills to enable the model to complete complex tasks; and (2) cross-embodiment learning to adapt the model to diverse robotic platforms.

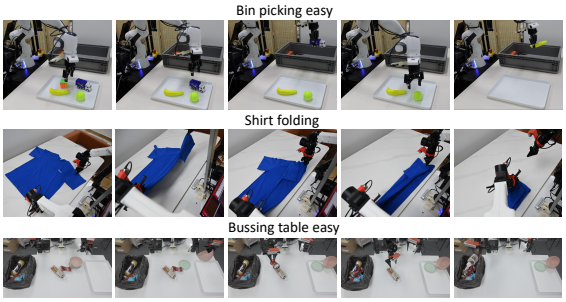

Figure 5: **Examples of tasks without task-specific adaptation.** We assessed our model's performance after stage 2 training using three tasks: **bin-picking easy** (top), **shirt folding** (middle), and **table bussing easy** (bottom).

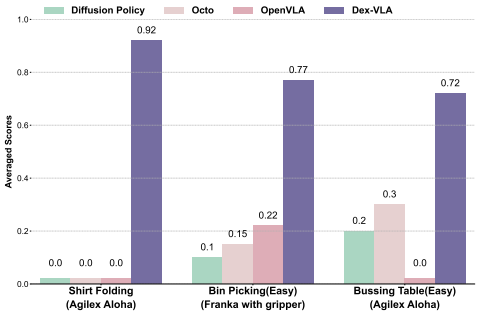

Figure 6: **Results on tasks without task-specific adaptation.** We compared our model against Octo, OpenVLA, and Diffusion Policy. Performance was evaluated across 10 trials for each model, with scores averaged across these trials.

Stage 1: Cross-embodiment pre-training. The vision-language-action models can be viewed as a composite of two distinct components. At the top of the architecture lies the vision-language model (VLM), which processes both visual input and language instructions, mapping them into a shared embedding space. This shared space is pre-trained using internet-scale data, enabling a wide range of capabilities, including language understanding, multimodal understanding, and various other vision-text tasks. However, despite its extensive training, the VLM lacks the ability to physically interact with diverse objects in real-world environments.

To effectively pre-train the action expert, we leverage all available data while temporarily decoupling it from the VLM component. This allows us to focus on developing a robust action generation capability independent of language grounding. We use a ResNet-50 as image encoders, aligning with DP [30] and DistilBERT [31] as a language embedding model. The resulting language embeddings are then integrated into the model using FiLM layers, consistent with previous work [32, 33].

Stage 2: Embodied-specific alignment. While Stage 1 learns basic motor skills from cross-embodied data, this cross-embodiment learning can potentially compromise performance on the target embodiment, making it unsuitable for real-world deployment. Stage 2 addresses this by training the model with embodiment-specific data, aligning the abstract vision-language representations from the VLM with the diffusion expert. Therefore, we filter the dataset to include only embodiment-specific data, ensuring each sample involves a single embodiment. Mirroring techniques employed in vision-language models like LLaVA [28, 34], this stage focuses on aligning the target embodiment's action space with its corresponding camera views and accompanying language instructions. Specifically, we jointly train the VLM model, the projection layer, and the diffusion expert on this embodiment-specific data, while freezing the VLM's visual encoder. This joint training allows the diffusion expert to effectively ground the high-level vision-language understanding from the VLM in the specific motor control space of the target robot. Following Stage 2 training, we observe that the model exhibits proficiency in performing a range of tasks on the target embodiment, such as shirt folding, demonstrating the effectiveness of the embodiment-specific training.

Stage 3: Task-specific adaptation. This phase refines the model's ability to execute downstream tasks skillfully and fluently, analogous to the post-training stage in large language models where the model is fine-tuned on domain-specific data. For simpler, less generalization-dependent tasks, such as shirt folding, table bussing, or bin picking with trained objects, task-specific training is unnecessary as the model already performs well. However, complex, dexterity-demanding tasks require the model to learn fine-grained, context-dependent actions. Therefore, effective post-training relies on a high-quality dataset of expert demonstrations exhibiting consistent and fluent task execution strategies focused on behaviors that promote successful task completion. It is worth noting that we utilize sub-step annotated language data in both Stage 2 and Stage 3. However, instead of directly using these sub-step reasoning as instructional input, we employ them as intermediate language output, compelling the model to learn and generate these sub-step language descriptions. This approach has proven highly effective, enabling our model to perform complex, long-horizon tasks such as laundry

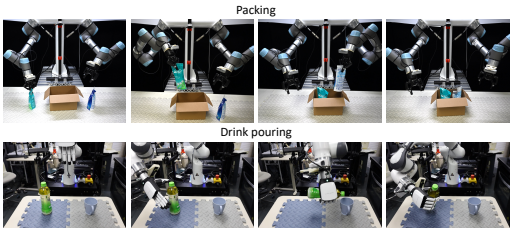

Figure 7: **Example of tasks for learning dexterous skills on new embodiment.** We evaluate our model on two new embodiments with **packing (top)** and **drink pouring (bottom)** tasks, which are not included in stage 1 & 2 train data.

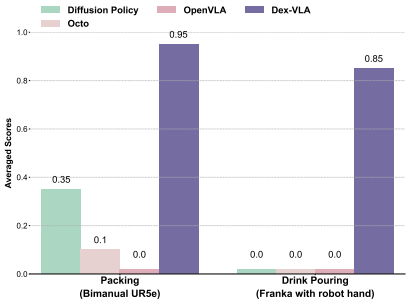

Figure 8: **Results on learning dexterous skills from new embodiment.** We evaluated our model with four baselines: Diffusion Policy, Octo, and OpenVLA. Diffusion Policy is directly trained on these novel tasks from scratch.

folding. While other VLAs, like π_0 [8], can also complete such tasks, they rely on high-level policy models like SayCan [35] to identify the task state and provide next-step instructions. In contrast, our framework leverages the VLM backbone as an implicit high-level policy model. This allows the model to internally interpret the task's state and inject this understanding into the policy to guide action generation, eliminating the need for an external high-level policy module.

Substep reasoning. A key insight for training the DexVLA is the necessity of decomposing long-horizon tasks (e.g., bussing tables) into sub-tasks. These tasks, often spanning beyond 2 minutes, prove challenging for the diffusion expert to learn effectively from a single language instruction. Therefore, we annotate sub-step instructions within these long-horizon tasks to provide a more structured learning signal. Pre-training with sub-steps is crucial for strong performance. We demonstrate the importance of sub-step reasoning in the Appendix. Our empirical observations show that VLA trained without this pre-training frequently skips critical steps in very long tasks. Sub-step annotations are typically provided every five seconds of the demonstration. We show some examples for substep reasoning in Figure 3.

3 Real World Experiments

3.1 Evaluating Model without Task-Specific Adaptation

This section evaluates the model's performance before task-specific adaptation (Stage 3). The evaluated tasks, visualized in Figure 5, all use the model with one set of parameters. Detailed task descrptions are listed in Appendix. These tasks vary significantly in trajectory length and complexity, with some requiring high dexterity and intricate manipulation (e.g., shirt folding). We benchmark our approach against OpenVLA [20], a 7B-parameter VLA model pre-trained on the Open X-Embodiment (OXE) [22] dataset, and Octo [21], a compact 93M parameter model that employs a diffusion-based policy for action generation. We used the open-sourced pre-trained weights for these two models. All baselines are fine-tuned on the same dataset and for the same number of epochs as our Stage 2 training, ensuring a fair comparison. We also compare to the Diffusion Policy [30], a strong baseline. Notably, neither Octo nor OpenVLA has previously demonstrated success on tasks of this complexity.

Following π_0, we use a normalized score averaged over 10 episodes per task and method as our evaluation metric. Detailed scoring rubrics for each task are provided in the Appendix. As shown in Figure 6, DexVLA significantly outperforms all baselines on all tasks without task-specific adaptation. Notably, baseline methods, including OpenVLA, Octo, and Diffusion Policy, struggled to complete any steps of the shirt folding task, highlighting its complexity. In contrast, DexVLA achieves a 0.92 point on shirt folding without any task-specific adaptation. A similar phenomenon is observed in the bin picking and table bussing tasks. While these challenging tasks sometimes see limited success from the baselines, their overall scores remain low. DexVLA, however, achieves substantially better performance on these tasks.

Figure 9: **Average scores on tasks requiring stage 3 training**. We compared our model against Octo, OpenVLA and π_0 on laundry folding and bussing table (hard).

Table 2: **Ablation study of three stage training**. To evaluate the impact of stage 1 training, we trained DexVLA from scratch without diffusion expert pre-training. Additionally, to assess the effectiveness of stage 3 training, we directly evaluated DexVLA on Laundry Folding without applying stage 3 training.

Stage 1	Stage 2	Stage 3	Shirt Folding	Laundry Folding
✓	-	-	0.0	0.0
-	✓	-	0.0	0.0
✓	✓	-	0.92	0.0
✓	✓	✓	0.92	0.4

Table 3: **Ablation results on size of Diffusion Expert.** We reported the average score on the shirt folding task.

DexVLA	UNet(93M)	Diff. Expert(410M)	Diff. Expert(1B)
Avg. Score	0.17	0.63	0.92

3.2 Learning Dexterous Skills on New Embodiment

This section evaluates our model's ability to learn dexterous skills on the new embodiments as shown in Figure 7. Specifically, the new embodiments are not involved in either Stage 1 or Stage 2 training. We aim to demonstrate the effectiveness of our proposed framework in acquiring new skills quickly on any new embodiment without the necessity of pre-training. Detailed task descrptions are listed in Appendix.

These tasks involve two novel robotic systems absent from training data: 1) **A Franka arm** integrated with a **dexterous hand** has 12 DoF, serving as a more complex robotic system than a simple gripper, 2) **A bimanual UR5e** system featuring humanoid-inspired kinematic design and its articulation fundamentally differs from conventional dual-arm platforms like the AgileX bimanual robot. We evaluate our approach against the same baselines as in the previous section. This section aims to validate the adaptability of our pre-trained model to new embodiments and tasks. To this end, we directly fine-tune our Stage 2 pre-trained model on the novel tasks. For OpenVLA and Octo, we employ their publicly available checkpoints pre-trained on the OXE dataset. The Diffusion Policy — a method specialized for learning dexterous tasks from limited data — is trained exclusively on the two novel tasks. All methods are trained on individual tasks, and to ensure fairness, each baseline undergoes the same number of training epochs.

Figure 8 compares the performance of the methods on two novel tasks. For each method and task, we report the averaged scores over 10 trials (detailed scoring criteria are provided in the Appendix). DexVLA achieves an average of 0.90 point across two tasks, while OpenVLA and Octo struggle. DexVLA significantly outperforms Diffusion Policy, achieving a substantial performance lead. These results highlight DexVLA's ability to efficiently adapt to new embodiments and master complex skills with only 100 demonstrations. These results are particularly meaningful as our method outperforms both extensively pre-trained VLA models (OpenVLA) and methods specifically designed for learning new tasks (Diffusion Policy).

3.3 Complex Long-Horizon Tasks with Direct Prompting

In this set of experiments, we tackle a range of challenging multi-stage tasks via a combination of task-specific post-training and self-generated reasoning capability. For some of these tasks, data is present in pre-training, but fine-tuning is required to attain mastery. For some, no data is present in pre-training. The tasks in this evaluation, are shown in Figure 3, and detailed task descrptions are listed in Appendix. These tasks involve extended-horizon challenges. For instance, laundry folding requires more than 2 minutes to collect a single episode, and the soft, deformable fabric of clothing generates numerous unseen shapes and states, posing significant challenges for recognition and task completion. In the sorting task, the model must pick up 5–8 randomly placed objects in a cluttered scene and relocate them to predefined target positions. We report two tasks, laundry folding and bussing table (hard), in the main text, and discuss the rest of three tasks in the Appendix.

Our evaluation of all models is based on averaged scores over 10 trials, with detailed scoring criteria provided in the Appendix. Conducting comparisons on these tasks is challenging due to the limited availability of prior models capable of operating at this scale. Consequently, we compare our method specifically against OpenVLA, Octo, and π_0. Note that π_0 is pretrained on extensive data and shares the same embodiment as our tested robot. We use direct prompting on all models for fair comparison.

The results, illustrated in Figure 11, demonstrate that DexVLA consistently outperforms all baseline methods. For the most complex task—laundry folding—our method achieves a score of 0.4, showcasing its potential to handle highly complicated scenarios. In comparison, π_0 attains a score of 0.2 when directly prompted with task instructions. On the table bussing task, DexVLA surpasses π_0 by 0.08 points. These findings suggest that DexVLA presents a promising solution for executing complex, long-horizon tasks without relying on an external high-level planner.

3.4 Ablation Study

Ablation on three-stage training strategy. To validate the necessity of this multi-stage training process, we conduct an ablation study in this section. Specifically, we evaluate the DexVLA model under several training conditions: training only Stage 1, only Stage 2, both Stage 1 and Stage 2 combined, and all three stages. Stage 1 consists of pretraining using cross-embodiment data, Stage 2 involves fine-tuning with embodiment-specific data, and Stage 3 enables the model to master more complex tasks. The experimental results are summarized in Table 2. We observe that training solely on Stage 1 or Stage 2 leads to a 0% success rate in folding tasks. Notably, the absence of Stage 1 training results in the model completely failing to learn any meaningful actions. We hypothesize that the considerable number of parameters in the diffusion expert complicates the optimization process. Thus, Stage 1 serves not only to equip the diffusion expert with foundational action skills but also to "warm up" its parameters, facilitating better comprehension of complex visual cues and language instructions. Additionally, to investigate the contribution of Stage 3 training to performance on more complex tasks, we directly evaluated DexVLA on the laundry-folding task without task-specific adaptation. We observed a significant performance drop from 0.4 to 0. These results highlight that Stage 3 training is essential for the model's success in handling long-horizon and challenging tasks.

Ablation on size of diffusion expert. Our key contribution is a novel vision-language-action (VLA) model architecture incorporating a diffusion expert, a significantly larger action expert based on a diffusion transformer. However, does the larger diffusion transformer architecture of the diffusion expert (1B) offer advantages over a smaller one?

To address this question, we utilize a tiny UNet-based diffusion policy (93M) and a smaller diffusion expert (410M) as baselines. As shown in Table 3, the UNet-based action expert performs significantly worse than our method, barely completing the shirt folding task with an average score of 0.17. Empirically, we observed oscillation in the robot's movements with the UNet model compared to our diffusion expert. We hypothesize that the UNet's fewer parameters contribute to interference between different actions in the parameter space, hindering the model's ability to learn the correct actions. The 410M diffusion expert achieved results lower than the 1B model (0.63 versus 0.92), indicating that learning a wide variety of tasks requires a greater number of model parameters.

4 Conclusion

This work proposes DexVLA, a novel architecture that leverages vision-language models to learn semantic information and employs a billion-parameter diffusion expert to learn robust and generalizable visuomotor policies. We introduce an embodied curriculum learning strategy, enabling the network to progressively learn from embodiment-agnostic motor skills to complex, embodiment-specific dexterous skills through three training stages. Furthermore, we incorporate sub-step reasoning, allowing the model to perform very long-horizon tasks without relying on a high-level policy model. Our method is evaluated from multiple perspectives, including its ability to perform complex tasks without task-specific adaptation, fine-tune on new embodiments with limited data, and execute extremely complex, long-horizon tasks without the assistance of a high-level policy model.

5 Limitations

Our method has several limitations and areas for improvement. First, we did not observe a significant transfer of learning from data collected on other embodiments to a specific target embodiment. Second, we observed that complex tasks, such as laundry folding, strongly depend on accurate action recovery and the ability to recognize the correct object state within the image. Prior work, π_0, addresses this by using SayCan, where a high-level policy model frequently assesses the object state and provides updated language instructions to the low-level vision-language-action model. Our current method, relying solely on the vision-language component, struggles with this aspect. A potential solution is to design a mechanism for explicit state checking and integrate it into our end-to-end model, which we plan to investigate in future work. Similar to observations reported in π_0, we also found that models trained with stage 3 exhibit reduced cross-task generalization compared to models trained solely through stages 1 and 2. However, complex tasks like laundry folding necessitate stage 3 training to effectively learn the required actions and object states for successful completion. Addressing this trade-off between task-specific learning and generalization is crucial for enabling models to perform complex tasks effectively without relying on post-training.

References

[1] T. Lin, Y. Zhang, Q. Li, H. Qi, B. Yi, S. Levine, and J. Malik. Learning visuotactile skills with two multifingered hands. *arXiv preprint arXiv:2404.16823*, 2024.

[2] H. Shi, H. Xu, S. Clarke, Y. Li, and J. Wu. Robocook: Long-horizon elasto-plastic object manipulation with diverse tools. *arXiv preprint arXiv:2306.14447*, 2023.

[3] F. Lin, Y. Hu, P. Sheng, C. Wen, J. You, and Y. Gao. Data scaling laws in imitation learning for robotic manipulation. *arXiv preprint arXiv:2410.18647*, 2024.

[4] K. Zhang, Z.-H. Yin, W. Ye, and Y. Gao. Learning manipulation skills through robot chain-of-thought with sparse failure guidance. *arXiv preprint arXiv:2405.13573*, 2024.

[5] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, et al. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. *The International Journal of Robotics Research*, 41(7): 690–705, 2022.

[6] Y. Qin, Y.-H. Wu, S. Liu, H. Jiang, R. Yang, Y. Fu, and X. Wang. Dexmv: Imitation learning for dexterous manipulation from human videos. In *European Conference on Computer Vision*, pages 570–587. Springer, 2022.

[7] M. Reuss, Ö. E. Yağmurlu, F. Wenzel, and R. Lioutikov. Multimodal diffusion transformer: Learning versatile behavior from multimodal goals. 2024.

[8] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2024. URL https://arxiv.org/abs/2410.24164.

[9] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[10] M. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[11] Y. Hu, F. Lin, T. Zhang, L. Yi, and Y. Gao. Look before you leap: Unveiling the power of gpt-4v in robotic vision-language planning. *arXiv preprint arXiv:2311.17842*, 2023.

[12] J. Liu, H. Chen, P. An, Z. Liu, R. Zhang, C. Gu, X. Li, Z. Guo, S. Chen, M. Liu, et al. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv preprint arXiv:2503.10631*, 2025.

[13] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, et al. pi0.5: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[14] M. J. Kim, C. Finn, and P. Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

[15] G. R. Team, S. Abeyruwan, J. Ainslie, J.-B. Alayrac, M. G. Arenas, T. Armstrong, A. Balakrishna, R. Baruch, M. Bauza, M. Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.

[16] J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[17] Q. Zhao, Y. Lu, M. J. Kim, Z. Fu, Z. Zhang, Y. Wu, Z. Li, Q. Ma, S. Han, C. Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. *arXiv preprint arXiv:2503.22020*, 2025.

[18] W. Zhao, P. Ding, M. Zhang, Z. Gong, S. Bai, H. Zhao, and D. Wang. Vlas: Vision-language-action model with speech instructions for customized robot manipulation. *arXiv preprint arXiv:2502.13508*, 2025.

[19] H. Zhen, X. Qiu, P. Chen, J. Yang, X. Yan, Y. Du, Y. Hong, and C. Gan. 3d-vla: A 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*, 2024.

[20] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, et al. Openvla: An open-source vision-language-action model.

[21] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[22] A. O'Neill, A. Rehman, A. Gupta, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

[23] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn. Humanplus: Humanoid shadowing and imitation from humans. *arXiv preprint arXiv:2406.10454*, 2024.

[24] H. Ha, Y. Gao, Z. Fu, J. Tan, and S. Song. Umi on legs: Making manipulation policies mobile with manipulation-centric whole-body controllers. *arXiv preprint arXiv:2407.10353*, 2024.

[25] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser. Tidybot: Personalized robot assistance with large language models. *Autonomous Robots*, 47(8):1087–1102, 2023.

[26] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang, et al. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11097–11107, 2020.

[27] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[28] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

[29] M. Zhu, Y. Zhu, J. Li, J. Wen, Z. Xu, N. Liu, R. Cheng, C. Shen, Y. Peng, F. Feng, et al. Scaling diffusion policy in transformer to 1 billion parameters for robotic manipulation. *arXiv preprint arXiv:2409.14411*, 2024.

[30] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

[31] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. arxiv 2019. *arXiv preprint arXiv:1910.01108*, 2019.

[32] L. X. Shi, Z. Hu, T. Z. Zhao, A. Sharma, K. Pertsch, J. Luo, S. Levine, and C. Finn. Yell at your robot: Improving on-the-fly from language corrections. *arXiv preprint arXiv:2403.12910*, 2024.

[33] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[34] H. Liu, C. Li, Y. Li, and Y. J. Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024.

[35] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

[36] H.-S. Fang, C. Wang, M. Gou, and C. Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11444–11453, 2020.

[37] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022.

[38] H. Ha, P. Florence, and S. Song. Scaling up and distilling down: Language-guided robot skill acquisition. In *Conference on Robot Learning*, pages 3766–3777. PMLR, 2023.

[39] I. Radosavovic, T. Xiao, S. James, P. Abbeel, J. Malik, and T. Darrell. Real-world robot learning with masked visual pre-training. In *Conference on Robot Learning*, pages 416–426. PMLR, 2023.

[40] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*, 2024.

[41] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[42] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess, S. Nair, Q. Vuong, O. Mees, C. Finn, and S. Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

[43] J. Wen, Y. Zhu, J. Li, M. Zhu, K. Wu, Z. Xu, N. Liu, R. Cheng, C. Shen, Y. Peng, et al. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2409.12514*, 2024.

[44] Z. Zhang, K. Zheng, Z. Chen, J. Jang, Y. Li, C. Wang, M. Ding, D. Fox, and H. Yao. Grape: Generalizing robot policy via preference alignment. *arXiv preprint arXiv:2411.19309*, 2024.

[45] Y. Guo, J. Zhang, X. Chen, X. Ji, Y.-J. Wang, Y. Hu, and J. Chen. Improving vision-language-action model with online reinforcement learning. *arXiv preprint arXiv:2501.16664*, 2025.

[46] S. Belkhale, T. Ding, T. Xiao, P. Sermanet, Q. Vuong, J. Tompson, Y. Chebotar, D. Dwibedi, and D. Sadigh. Rt-h: Action hierarchies using language. *arXiv preprint arXiv:2403.01823*, 2024.

[47] L. Yen-Chen, A. Zeng, S. Song, P. Isola, and T.-Y. Lin. Learning to see before learning to act: Visual pre-training for manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7286–7293. IEEE, 2020.

[48] Y. Du, M. Simchowitz, R. Tedrake, V. Sitzmann, B. Chen, and D. M. Monso. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *NeurIPS*, 3, 2024.

[49] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[50] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[51] T. Z. Zhao, J. Tompson, D. Driess, P. Florence, S. K. S. Ghasemipour, C. Finn, and A. Wahid. Aloha unleashed: A simple recipe for robot dexterity. In *8th Annual Conference on Robot Learning*.

[52] Y. Wang, Y. Zhang, M. Huo, R. Tian, X. Zhang, Y. Xie, C. Xu, P. Ji, W. Zhan, M. Ding, et al. Sparse diffusion policy: A sparse, reusable, and flexible policy for robot learning. *arXiv preprint arXiv:2407.01531*, 2024.

[53] A. Prasad, K. Lin, J. Wu, L. Zhou, and J. Bohg. Consistency policy: Accelerated visuomotor policies via consistency distillation. *arXiv preprint arXiv:2405.07503*, 2024.

[54] M. Uehara, Y. Zhao, K. Black, E. Hajiramezanali, G. Scalia, N. L. Diamant, A. M. Tseng, T. Biancalani, and S. Levine. Fine-tuning of continuous-time diffusion models as entropy-regularized control. *arXiv preprint arXiv:2402.15194*, 2024.

[55] M. Uehara, Y. Zhao, K. Black, E. Hajiramezanali, G. Scalia, N. L. Diamant, A. M. Tseng, S. Levine, and T. Biancalani. Feedback efficient online fine-tuning of diffusion models. *arXiv preprint arXiv:2402.16359*, 2024.

[56] K. Black, M. Janner, Y. Du, I. Kostrikov, and S. Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

[57] K. Black, M. Nakamoto, P. Atreya, H. Walke, C. Finn, A. Kumar, and S. Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *arXiv preprint arXiv:2310.10639*, 2023.

[58] S. Dasari, O. Mees, S. Zhao, M. K. Srirama, and S. Levine. The ingredients for robotic diffusion transformers. *arXiv preprint arXiv:2410.10088*, 2024.

[59] F. Lin, Y. Hu, P. Sheng, C. Wen, J. You, and Y. Gao. Data scaling laws in imitation learning for robotic manipulation, 2024. URL https://arxiv.org/abs/2410.18647.

[60] A. Z. Ren, J. Lidard, L. L. Ankile, A. Simeonov, P. Agrawal, A. Majumdar, B. Burchfiel, H. Dai, and M. Simchowitz. Diffusion policy policy optimization. *arXiv preprint arXiv:2409.00588*, 2024.

[61] Y. Wang, L. Wang, Y. Du, B. Sundaralingam, X. Yang, Y.-W. Chao, C. Perez-D'Arpino, D. Fox, and J. Shah. Inference-time policy steering through human interactions. *arXiv preprint arXiv:2411.16627*, 2024.

[62] N. Liu, S. Li, Y. Du, A. Torralba, and J. B. Tenenbaum. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, pages 423–439. Springer, 2022.

[63] Y. Hu, Y. Guo, P. Wang, X. Chen, Y.-J. Wang, J. Zhang, K. Sreenath, C. Lu, and J. Chen. Video prediction policy: A generalist robot policy with predictive visual representations. *arXiv preprint arXiv:2412.14803*, 2024.

[64] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *arXiv preprint arXiv:2402.10885*, 2024.

[65] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *ICRA 2024 Workshop on 3D Visual Representations for Robot Manipulation*, 2024.

[66] Y. Ze, Z. Chen, W. Wang, T. Chen, X. He, Y. Yuan, X. B. Peng, and J. Wu. Generalizable humanoid manipulation with improved 3d diffusion policies. *arXiv preprint arXiv:2410.10803*, 2024.

[67] G. Yan, Y.-H. Wu, and X. Wang. Dnact: Diffusion guided multi-task 3d policy learning. *arXiv preprint arXiv:2403.04115*, 2024.

[68] X. Jia, Q. Wang, A. Donat, B. Xing, G. Li, H. Zhou, O. Celik, D. Blessing, R. Lioutikov, and G. Neumann. Mail: Improving imitation learning with selective state space models. In *8th Annual Conference on Robot Learning*.

[69] J. Wen, M. Zhu, Y. Zhu, Z. Tang, J. Li, Z. Zhou, C. Li, X. Liu, Y. Peng, C. Shen, et al. Diffusion-vla: Scaling robot foundation models via unified diffusion and autoregression. *arXiv preprint arXiv:2412.03293*, 2024.

[70] K. Wu, Y. Zhu, J. Li, J. Wen, N. Liu, Z. Xu, Q. Qiu, and J. Tang. Discrete policy: Learning disentangled action space for multi-task robotic manipulation. *arXiv preprint arXiv:2409.18707*, 2024.

[71] L. Wang, K. Zhang, A. Zhou, M. Simchowitz, and R. Tedrake. Fleet policy learning via weight merging and an application to robotic tool-use. *arXiv preprint arXiv:2310.01362*, 2023.

[72] L. Wang, J. Zhao, Y. Du, E. H. Adelson, and R. Tedrake. Poco: Policy composition from and for heterogeneous robot learning. *arXiv preprint arXiv:2402.02511*, 2024.

[73] L. Wang, X. Chen, J. Zhao, and K. He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. *arXiv preprint arXiv:2409.20537*, 2024.

A Related Work

Vision-Language-Action models for robot control. Recent research has focused on developing generalist robot policies trained on increasingly expansive robot learning datasets [36, 37, 23, 24, 38, 3, 39, 40, 41]. Vision-language-action models (VLA) [20, 8, 42, 43, 9, 19, 44, 45, 46] represent a promising approach for training such generalist policies. VLAs adapt vision-language models, pre-trained on vast internet-scale image and text data, for robotic control [47]. This approach offers several advantages: leveraging large vision-language model backbones, with billions of parameters, provides the necessary capacity for fitting extensive robot datasets. Furthermore, reusing weights pre-trained on internet-scale data enhances the ability of VLAs to interpret diverse language commands and generalize to novel objects and environments. However, current VLA models do not specifically focus on learning dexterous robotic skills by leveraging the parameters of the underlying VLM. While a few works, such as π_0 [8] and TinyVLA [43], introduce external action experts to facilitate action learning, their training pipelines still rely on the entire model. Another challenge is that even advanced methods like π_0, despite being capable of completing highly dexterous and long-horizon tasks, require the assistance of a high-level policy, such as SayCan [35], to decompose tasks into sub-goals. This allows the VLA to complete sub-tasks sequentially. We aim to integrate this high-level planning capability directly into the model itself by training each component of the network with data annotated at the sub-step level. Consequently, our method can complete complex tasks, like laundry folding, without requiring an external high-level policy, making the entire framework more end-to-end and demonstrating significant potential.

Diffusion models. Diffusion models [48, 49, 50] have emerged as the dominant approach in visual generation. The Diffusion Policy [30] successfully applies the diffusion model to robot learning, demonstrating its ability to model multimodal action distributions. Subsequent research has further developed the Diffusion Policy [51, 52, 53, 7, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63] by applying it to 3D environments [64, 65, 66, 67], scaling its capabilities [29], improving its efficiency [68, 53], and incorporating architectural innovations. There are a number of works investigating the usage of diffusion VLA [43, 8, 69]. Although existing models achieve strong performance and generalization on diverse tasks, they predominantly rely on the capabilities of pre-trained vision-language models. This work proposes a paradigm shift towards the diffusion module, demonstrating that a newly designed diffusion-based action expert, coupled with a novel training strategy, enables VLA models to learn from data more efficiently and effectively.

B More Experimental Results

B.1 Visual generalization.

Visual generalization is a critical aspect of robot learning. A well-trained model should not only perform well on in-domain tasks but also generalize to different objects within the same category and to novel scenes. This section presents our visual generalization tests. Specifically, we evaluate shirt folding on a bimanual AgileX and drink pouring using a Franka Emika robot with a dexterous hand. The former task is evaluated without task-specific adaptation, while the latter is trained with 100 demonstrations of the new embodiment. These tasks were also the focus of the experiments presented in Section 3.1 and Section 3.2, respectively. For both tasks, we assess visual generalization across two dimensions: novel objects and novel scenes.

For shirt folding, we varied the shirt color (while maintaining size) and altered the background and scene. For drink pouring, we used unseen cups and bottles, also evaluating the task in different scenes and backgrounds. The results are presented in Table 4. Our experiments demonstrate that DexVLA effectively generalizes to novel visual environments. As shown in the supplementary video, the model successfully handles even challenging cases, such as folding white shirts on a white table. The examples are shown in Figure 10.

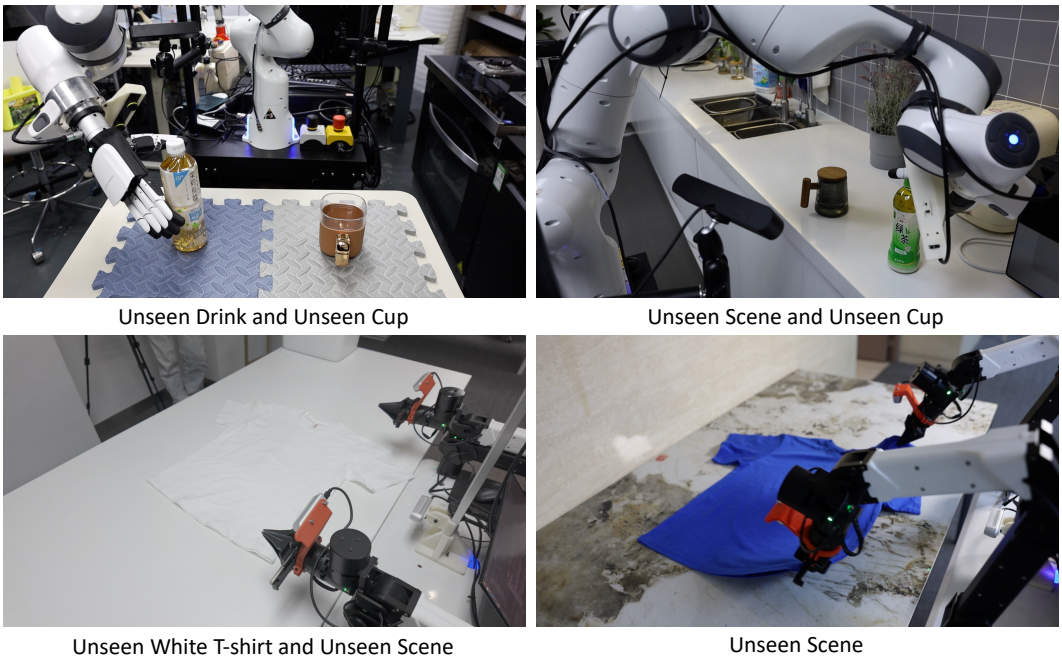

Figure 10: **Example of visual generalization.** Here lists some visual generalization settings including unseen objects and unseen scenes.

Table 4: **Visual Generalizaton for DexVLA**. For each evaluation setting, we report the averaged scores across 3 trials.

Task / Generalization	Embodiment	Novel Object	Novel Scene	Novel Object & Scene
Shirt folding	Bimanual AgileX	0.78	0.78	0.56
Drink pouring	Dexterous hand	0.83	0.67	0.67

B.2 LIBERO simulation experimental results.

We compare our method on the Libero benchmark. DexVLA uses Stage 1 pre-trained weights. Specifically, the experimental results in Table 5 show that DexVLA outperforms all baselines, including π_0 and π_0-FAST, two state-of-the-art VLA methods, demonstrating strong performance on the benchmark.

B.3 Training cost of stage 1.

As mentioned in Section 2.2, training the entire VLA model from scratch results in failure on nearly all tasks. Therefore, this section compares the training cost of our Stage 1 (training only the diffusion expert) with that of training the entire VLA model. The test reports the number of training epochs completed per hour. We deliberately keep the same batch size for fair comparison.

As shown in Table 6, training only the diffusion expert is 2.78 times faster than training the entire VLA model. This is expected, as the VLA model is three times larger than the diffusion expert alone. This highlights that our training strategy is not only effective but also cost-efficient.

B.4 More discussion on long-horizon tasks with direct prompting.

In 3.3, we previously compared DexVLA with π_0, Octo, and OpenVLA on laundry folding and bussing table hard. Here, we present additional results against OpenVLA and Octo on more complex tasks. As shown in Figure 11, DexVLA consistently outperforms all baselines, achieving 0.8

Table 5: **Evaluations results on LIBERO.** We compare our DexVLA with DP and OpenVLA.

Method	Spatial	Object	Goal	Average
DP	78.3	92.5	68.3	79.7
OpenVLA	84.7	88.4	79.2	84.1
π_0-FAST	96.4	96.8	88.6	93.9
π_0	96.8	98.8	**95.8**	97.1
DexVLA	**97.2**	**99.1**	95.6	**97.3**

Table 6: **Comparison of training cost for train only diffusion expert versus train entire VLA.** Training cost is measured by the number of training epochs completed per hour.

Train Method	Train only Diffusion Expert	Train Entire VLA
Training Cost	0.89 epoch / hour	0.32 epoch / hour

points in the dryer unloading task, while Octo and OpenVLA score 0 points. In tasks like sorting and bin picking hard, DexVLA achieves a score nearly 3 times higher than other baselines. Overall, DexVLA demonstrates its versatility in handling complex tasks through our embodied curriculum learning method, scalable diffusion expert, and novel VLA framework. We believe DexVLA presents a promising approach for building VLA systems capable of managing heterogeneous robotics data and mastering intricate manipulation tasks.

B.5 Zero-Shot Cross-Embodiment Transfer

Finally, we pose an intriguing question: can DexVLA perform zero-shot cross-embodiment transfer? To explore this, we take a model trained on a simple two-finger gripper and deploy it, without any further training, on a dexterous five-finger hand. Specifically, we use the Stage 2 pre-trained DexVLA — which was trained on a Franka robot equipped with a Robotiq gripper — and swap in a dexterous hand at test time. Because the gripper has only one degree of freedom, we constrain the dexterous hand to a single degree of freedom as well.

We evaluate on a bin-picking task with 30 novel objects that were unseen during both Stage 1 and Stage 2 training. An illustrative example appears in Figure 12, and a full video demonstration is provided in the supplementary material. Across these 30 objects, we achieve an average success rate of 60%. While this is slightly below the 67% success rate attained with the original gripper, it nonetheless underscores the model's robustness. The performance gap arises from three main challenges: (1) the dexterous hand's appearance differs markedly from the gripper, forcing the model to generalize its visual feature representations; (2) the wrist camera's mounting position shifts significantly, requiring adaptation to a new viewpoint; and (3) the difference in hand height changes the effective object grasping height. Although we do not yet control all degrees of freedom needed for fully dexterous manipulation, these results demonstrate that DexVLA's visual and camera-view representations transfer effectively across embodiments.

C Ablation Study

C.1 Does training with substep reasoning help?

A key strength of our method is its ability to handle extremely long and complex tasks, such as folding randomly crumpled shirts from a basket. It also enables the model to complete multi-stage tasks like shirt folding and bin picking without requiring post-training. Therefore, we now examine the importance of sub-step reasoning. We conducted an ablation study with two setups: 1) The diffusion expert is trained with direct prompting (each task has only one language instruction), while the VLA-diffusion expert is trained with sub-step reasoning. 2) Both stage 1 and stage 2 are trained

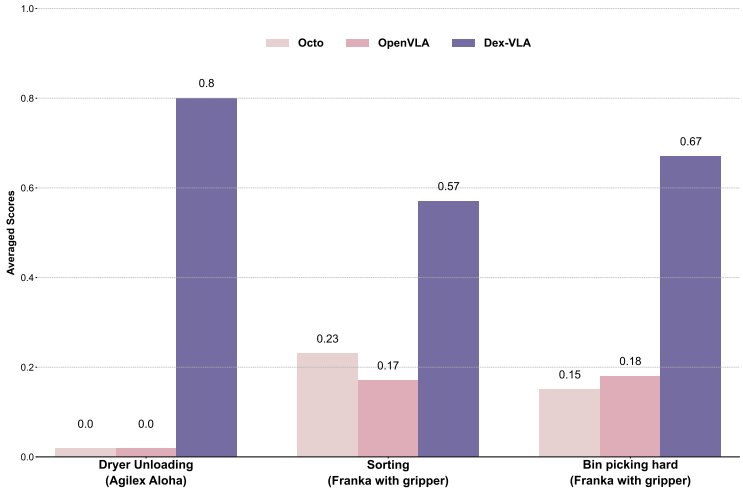

Figure 11: **Average scores on tasks requiring stage 3 training**. We compared our model against two baselines: Octo and OpenVLA. Averaging scores over 10 trials, our method significantly outperformed both baselines across all tasks. Note that sorting was not included in the pre-training data.

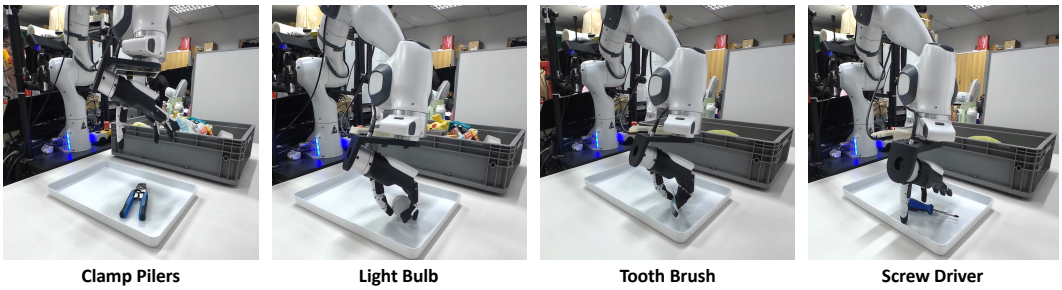

Figure 12: **Robot setup on zero-shot cross-embodiment transfer from gripper to dexterous hand.** We replace the original gripper with an Inspire dexterous hand and evaluate DexVLA in the same bin-picking environment using unseen objects.

with direct prompting data. The results are shown in Table 7. Training the diffusion expert with direct prompting, even for a relatively simple task like shirt folding, reduces the averaged score from 0.92 to 0.07. Furthermore, removing sub-step reasoning from both stages results in a complete failure (0 score). This is a significant observation. It suggests that learning long-horizon tasks within a shared parameter space can sometimes lead to conflicts. We hypothesize that sub-step reasoning allows the model to learn a more disentangled action space, similar to mapping a continuous action space to a discrete one [70]. This effectively segments the shared parameter space, allocating a smaller set of parameters to each substep [71, 72, 73]. This avoids parameter conflicts, leading to improved performance and generalization.

C.2 Explicit versus implicit sub-step reasoning.

The π_0 utilizes the additional SayCan to explicitly plan substeps, while DexVLA generates substep reasoning implicitly. Here, we evaluate the impact of these two reasoning approaches on overall performance. Specifically, we replace DexVLA's implicit substep reasoning with SayCan's. As shown in Table 8, models using implicit substep reasoning significantly outperform those relying on SayCan. The key advantage of implicit substep reasoning is its ability to disentangle the action space across different features, allowing DexVLA to learn more effectively. Additionally, SayCan updates instructions at a fixed frequency (every two seconds), which can result in redundant or repeated

Table 7: **Ablation study of substep reasoning.** The ✓in each stage indicates the use of sub-step reasoning data during that stage. We report the average score on the shirt-folding task.

Stage 1	Stage 2	Averaged Score
	✓	0.07
✓		0
✓	✓	0.92

Table 8: **SayCan versus Substep Reasoning for DexVLA.** We replace DexVLA's substep reasoning with SayCan and evaluate how this change affects overall performance.

Tasks/Models	SayCan	Substep Reasoning
Bussing Table (Hard)	0.58	0.70

states in certain scenarios. In contrast, DexVLA's substep reasoning adaptively segments the state space over the course of long-horizon tasks, contributing to its superior performance.

D Task Suite and Evaluation Protocol

D.1 Task suite.

These tasks are evaluated in Section 3.1.

- **Shirt folding (Bimanual AgileX)**: The shirt is placed flattened on the table, and the robot is asked to fold a t-shirt. We evaluate two shirts, a yellow shirt of medium size and a blue shirt of large size.
- **Bin picking easy (Franka with gripper)**: The model needs to pick up all items from the right panel to the left tray. All items are seen in the dataset.
- **Bussing table easy (Bimanual AgileX)**: The robot must clean a table, place dishes and cutlery in a bin, and trash into a trash bin.

These tasks are evaluated in Section 3.2.

- **Drink pouring (Franka with dexterous hand)**: The drink is placed on the right of the table and a cup is placed on the left. The robot needs to grab the drink and pour it into the cup. This task includes 100 demonstrations.
- **Packing (Bimanual UR5)**: The robot is asked to pick up objects on both sides and place them into the box for packing. This task includes 100 demonstrations.

These tasks are evaluated in Section 3.3.

- **Laundry folding (Bimanual AgileX)**: This task requires a static (non-mobile) bimanual system to fold articles of clothing. The clothing items start in a randomized crumpled state in a bin, and the goal is to take out the item, fold it, and place it on top of a stack of previously folded items. The randomized initial configuration of the crumpled laundry presents a major challenge since the policy needs to generalize to any configuration. This task is present in pre-training.
- **Dryer unloading (Bimanual AgileX)**: Here, the AgileX mobile robot has to take the laundry out of a dryer and place it into a hamper. This task is present in pre-training.
- **Sorting (Franka with gripper)**: The model needs to pick up all items from the right panel, and place them in the correct subsection of the left tray. The left tray is divided into four subsections. All new objects belong to the same category as these four sections. This task includes 200 demonstrations. This task is not in the pre-training data.

- **Bin picking hard (Franka with gripper)**: The model needs to pick up all items from the right panel to the left tray. Unlike the easy version, all objects are new and only present at test time. This task is present in pre-training.
- **Bussing table hard (Bimanual AgileX)**: The robot must clean a table, place dishes and cutlery in a bin, and trash into a trash bin. Unlike the easy version, all objects are new. In particular, we use dishes with unseen colors and trash with different appearances. This task is present in pre-training.

D.2 Evaluation protocol.

Each task is evaluated across 10 trials and reported averaged scores. For each task, we list the detailed scoring criterion as follows.

- **Lanudary folding (Bimanual AgileX)**: This task is scored out of 4 and we evaluate 5 shirts in total including 2 middle size and 3 small size. We perform two trials for each item, and the items left to be evaluated starting randomly crumpled in a laundry bin (while previously evaluated items start in a fold). One point is given for picking an item out of the bin and putting it on the table. Another point is given for flattening the shirt or shorts. A third point is granted for folding the shirt or shorts. A final point is given for either placing the item in the corner of the table (if it is the first item evaluated), or stacking it onto an existing stack of folded clothes. This evaluation metric is followed π_0.
- **Shirt folding (Bimanual AgileX)**: This task is scored out of 3. We perform two trials for each item, and the items are flattened on the table. One point is given for double vertically fold. Another point is granted for a double horizontal fold. A final point is given for pushing the folded shirt to the right blank area.
- **Bussing table (Bimanual AgileX)**: This task is scored out of 3-4 where there are 3-4 objects on the table in both **easy and hard version**. The main difference is the objects that appeared in the hard version are unseen. A point is given for each correctly sorted object.
- **Dryer unloading (Bimanual AgileX)**: This task is scored out of 2 where there are 2 crumpled shirts in the dryer. A point is given for pick up a shirt and place into the hamper.
- **Sorting (Franka with gripper)**: This task is scored out of 5-8 where there are 5-8 objects on the table. There are four kinds of objects in total, a point is given for each correctly sorted object.
- **Drink pouring (Franka with dexterous hand)**: This task is scored out of 2. A point is given for grab the bottle and pour to the cup. Another point is granted for place down the bottle.
- **Bining picking (Franka with gripper)**: This task is scored out of 4-5 where there are 4-5 objects on the table. The main difference is the objects that appeared in the hard version are unseen. A point is given for each correctly picked and placed object.
- **Packing (Bimanual UR5e)**: This task is scored out of 2 where there are 2 objects on the table. A point is given for each correctly picked and placed object.

E More Implementation Details

E.1 Robot setup.

Our evaluation is conducted on four different robot configurations across 10 tasks. These setups are summarized in the following list and visualized in figure 4.

- **Franka with gripper.** A Franka Emika robot with 7 degrees of freedom, equipped with a Robotiq parallel jaw gripper. Data is collected at 15Hz.

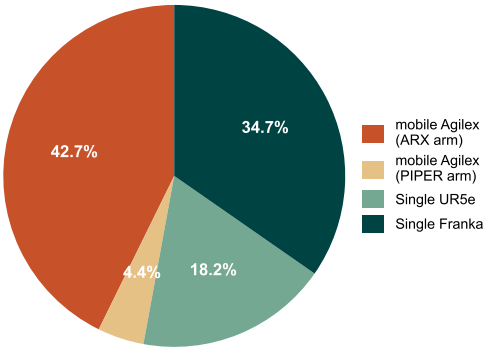

Figure 13: Overview of our dataset for stage 1 training.

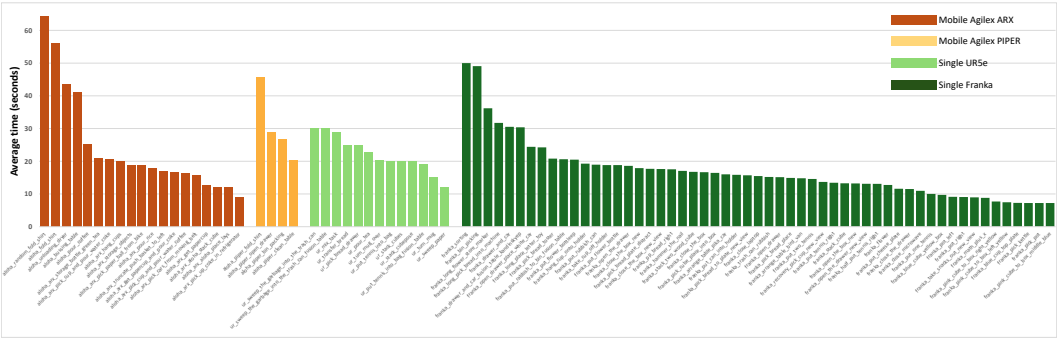

Figure 14: Average task length(seconds) of 91 tasks.

- **Franka with dexterous hand.** An Inspired Dexterous Hand mounted on a Franka Emika robot. The camera setup is identical to the gripper version, with a total of 12-dimensional configuration. Specifically, SE(3) end-effector pose (3D position + 3D orientation) plus 6-dimensional hand joint space. Data is collected at 15Hz.
- **Bimanual UR5e.** Two UR5e robots, each with a Robotiq parallel jaw gripper and a wrist-mounted camera. A top-down camera is positioned between the two arms. This setup has a total of three camera views and a 14-dimensional configuration and action space. Data is collected at 10Hz.
- **Bimanual AgileX.** Two 6-DoF AgileX arms, each with a wrist-mounted camera and a base camera. This setup has a 14-dimensional configuration and action space, supported by three cameras in total. Data is collected at 10Hz.

Our setup includes two Franka Emika robots: one equipped with a gripper and the other with a robot hand. Both Franka robots utilize the same camera configuration, consisting of a ZED 2 camera positioned on both the left and right sides, as well as a ZED Mini wrist camera mounted on the robot itself. Our bimanual UR5e robot uses a single top-mounted Intel RealSense L515 camera and two Intel RealSense 435i cameras attached to the wrists. Finally, our mobile AgileX platform has two Intel RealSense 435i wrist cameras and a top-mounted Intel RealSense 457 camera. Although the mobile AgileX image includes a front camera, it was not used during either training or inference.

E.2 Sub-step reasoning and data acquisition.

Training with sub-step reasoning is crucial for Dex-VLA to complete long-horizon tasks without a high-level policy model. We present an ablation study on the importance of sub-step reasoning in Section C.1. Acquiring this data presents two key challenges: obtaining language instructions and

segmenting videos with corresponding annotations. We address these challenges with the following strategy.

For object-level tasks (e.g., bin picking, sorting, table bussing), object identification is key. We leverage Grounding-Dino and DINOv2 to annotate object bounding boxes and names, along with the gripper's bounding box. We then calculate the intersection over union between the gripper and object bounding boxes to determine grasp success. For long-horizon single-object tasks (e.g., fold one shirt), the challenge lies in task segmentation. We created a comprehensive list of potential sub-step reasoning, focusing on major steps lasting at least five seconds each to avoid excessive sub-division. We then used Google Gemini 2.0, providing it with the sub-step list, to segment the videos and select the corresponding reasoning from the list. This proved effective and efficient for labeling. We only manually checked the Stage 3 training data, as this stage requires higher-quality annotations. This annotation strategy makes our approach feasible.

E.3 Architectural details.

In this section, we provide a full description of the model architecture. Dex-VLA can be split into two parts, VLM backbone originates from Qwen2-VL [27] and diffusion expert. We use Qwen2-VL 2B which is powerful and efficient. Regarding our Diffusion Expert, the total number of parameters for this model is 1 billion parameters. We use 32 layers, with the hidden stage of 1280, and a number of heads of 16. During Stage 1, we only pre-train the diffusion expert with random initialized ResNet-50 to process images and off-the-shelf Distilbert [31] to encode language instructions. Because the original diffusion policy model does not support cross-embodiment training, we adopted a multi-head structure similar to Octo [21]. Each embodiment is assigned a unique MLP head. The diffusion expert is trained using the similar settings of our Dex-VLA. In particular, we use the image resolution of 320×240, with three camera views. Each image is processed independently to a ResNet-50. We use the strategy as in RT-1 [33] to initialize the FiLM layers.

E.4 Training data details.

As shown in Figure 13. Our dataset comprises approximately 100 hours of collected data spanning 91 distinct tasks. The majority of this data was collected using two robot platforms: the Agilex (ARX arm) (42.7%) and the single Franka Emika robot (34.7%). The "ARX arm" and "PIPER arm" represent two distinct robotic arm configurations, both featuring six degrees of freedom (6-DoF) but differing in their kinematic structures and operational characteristics.

Figure 14 provides a summary of all 91 pretraining tasks across four embodiments. The Y-axis represents the average duration (in seconds) recorded for each task in the training data, while the X-axis lists all 91 tasks. It is clear that most tasks in the dataset are short-horizon, whereas the evaluated tasks are long-horizon, highlighting a notable distributional difference between the pretraining data and the evaluated tasks.