UNICOD: ENHANCING ROBOT POLICY VIA UNIFIED CONTINUOUS AND DISCRETE REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Joint state prediction has garnered significant attention in vision-language-action models. However, prior approaches have predominantly focused on visual signal generation, while overlooking multimodal understanding and the dynamic characterization of high-dimensional features. Recent unified models of generation and understanding have demonstrated strong capabilities in both comprehension and generation through large-scale pretraining. We posit that robotic policy learning can likewise benefit from the combined strengths of observation understanding and continuous future representation learning. Building on this insight, we introduce XXX, which acquires the ability to dynamically model high-dimensional visual features through pretraining on over 1M internet-scale instructional manipulation videos. Subsequently, XXX is fine-tuned on data collected from the robot embodiment, enabling the learning of mappings from future state representations to action tokens. Extensive experiments show our approach consistently outperforms baseline methods in terms of xxx and xxx across both simulation environments and real-world tasks.

1 INTRODUCTION

Constructing generalist foundation models for robots operating in the physical world has emerged as a rapidly growing frontier within embodied AI research. Vision–language–action (VLA) models aim to learn robotic policies from datasets annotated with perception, language, and action signals. However, the scarcity of robotic data and the heterogeneity across embodiments present substantial challenges, particularly in achieving generalization to novel scenes and task instructions.

To mitigate these limitations, recent studies have explored mapping vision–language models (VLMs), pretrained on large-scale vision–language data, into the action space. This strategy provides VLAs with alignment priors across language and vision modalities. Nevertheless, these approaches often overlook the fundamental discrepancies between robotic action tasks and vision–language tasks. Unlike the abundance of internet-scale vision–language data, fine-tuning VLMs on limited robotic datasets frequently leads to degradation of their foundational capabilities. Complementary lines of work have investigated leveraging image or video generation models as intermediaries for action policy learning. While such visual foresight approaches facilitate dynamic representation learning and enable the use of heterogeneous data sources, they typically fail to preserve the strong vision–language alignment inherent to pretrained VLMs. These observations highlight a central insight: it is crucial to design task-specific post-training paradigms tailored to embodied scenarios.

In parallel, unified modeling approaches that jointly address perception and future state prediction in multimodal contexts have advanced rapidly. These methods integrate heterogeneous tasks and exhibit emergent capabilities as scale increases. But the focus has seldom extended to learning agent actions and behaviors. However, upon re-examining this line of approaches, we observe that both language understanding and future state prediction can provide preliminary guidance for general manipulation tasks. This unified learning strategy further enables the model to acquire representations beneficial for robotic tasks from a broader range of data.

Building upon these insights and prior advances in vision–language–action (VLA) research, we propose UniCoD, a unified framework for robot learning. UniCoD follows an understand-

ing–generation–execution paradigm that integrates discrete task comprehension with continuous prediction of future robotic states. To address heterogeneous modalities, UniCoD employs a Mix-of-Transformers architecture with modality-specialized experts. To preserve the vision–language capabilities of the model while enabling action learning to benefit from both instruction understanding and future state forecasting, UniCoD is trained in two stages:

In the first stage, we curate a diverse collection of cross-embodiment manipulation data from both robots and human demonstrations. We leverage vision–language models (VLMs), guided by prompt-based annotations, to obtain language signals related to robotic instruction planning, and employ vision models to extract predictive visual features. This stage is exclusively dedicated to training on vision–language forecasting without applying supervision over the action space.In the second stage, we introduce embodiment-specific robotic data annotated with action behaviors. Building upon the pretrained representations from the first stage, the model learns to adapt visual and language features to the embodiment, and maps them into the action space through an action expert. Furthermore, we assess the effectiveness of different vision models by comparing their representations under manipulation accuracy metrics.

We introduce UniCoD, a general vision-language-action pretrained framework that unifies continuous and discrete representations. In experiments, UniCoD compares with other baselines across two simulated and two real-world embodiments. Notably, the UniCoD achieves a xxx improvement in the Simpler benchmark compared to existing SOTA approach. UniCoD also demonstrates higher success rates on real-world robots for complex tasks, particularly those requiring generalization to novel instructions. In summary, our contributions are as follows:

• We propose a unified vision–language–action (VLA) pretraining framework that integrates both discrete and continuous representations. The model is pretrained on the diverse dataset we collected and processed, enabling effective transfer to embodied tasks.

• We propose a two-stage training strategy for vision–language–action tasks that aligns action representations while preserving the capabilities of the original pretrained model.

• Our best-performing model achieves state-of-the-art results across both simulated and real-world environments, and we further analyze the impact of different feature design choices on the model's capabilities.

2 Related Works

Vision-Language-Action Models Vision-Language-Action (VLA) models introduce multimodal large language models (Dai et al., 2024; Touvron et al., 2023; Wang et al., 2025a; Bai et al., 2025) into robot policy models to enhance their generalization ability (Brohan et al., 2023; Kim et al., 2024; Black et al., 2024; Guo et al., 2025). This line of work either utilizes the VLM and an action head for end-to-end action prediction (Li et al., 2023; Wen et al., 2025) or uses the VLM to extract key information to condition downstream policy (Zhang et al., 2024; Li et al., 2025). Some recent works have introduced additional auxiliary tasks to VLAs, including enhancing spatial understanding (Qu et al., 2025), QA reasoning (Zhou et al., 2025), visual reasoning (Zhao et al., 2025) and prediction (Zhang et al., 2025), demonstrating that both general-purpose understanding and generation capabilities can promote action learning. However, these methods are primarily limited to unifying generative tasks within a discrete token prediction framework, which may compromise the robust vision-language alignment inherent in the pre-trained VLM. In this work, while utilizing discrete language prediction tasks to maintain the VLM's intrinsic feature alignment, we also incorporate a continuous-space visual prediction task to aid downstream action learning.

Generalist Robot Policies with Joint Prediction Explorations into generalist robot policies have considered using world models (Blattmann et al., 2023; Assran et al., 2025) to learn physical dynamics and subsequently predict actions (Du et al., 2024; Black et al., 2023). Many recent methods have incorporated prediction into larger-scale data and models: GR-1 (Wu et al., 2023) utilizes video pre-training to initialize the action policy; PAD (Guo et al., 2024) attempts to simultaneously decode future images and actions using diffusion policy; and VPP (Hu et al., 2024) uses a video foundation model as the visual encoder for action policy. While these methods fully leverage the rich information from video data, they lack semantic grounding capabilities due to the absence of large language

models. Recent works (Zhang et al., 2025; Wang et al., 2025b) use VQ quantization to incorporate predictive generation tasks into VLA policies, demonstrating the potential for unifying understanding and prediction. In contrast, we utilize continuous visual features as the prediction supervision signal and pre-train our model on large-scale language prediction and continuous visual prediction tasks.

3 METHODOLOGY

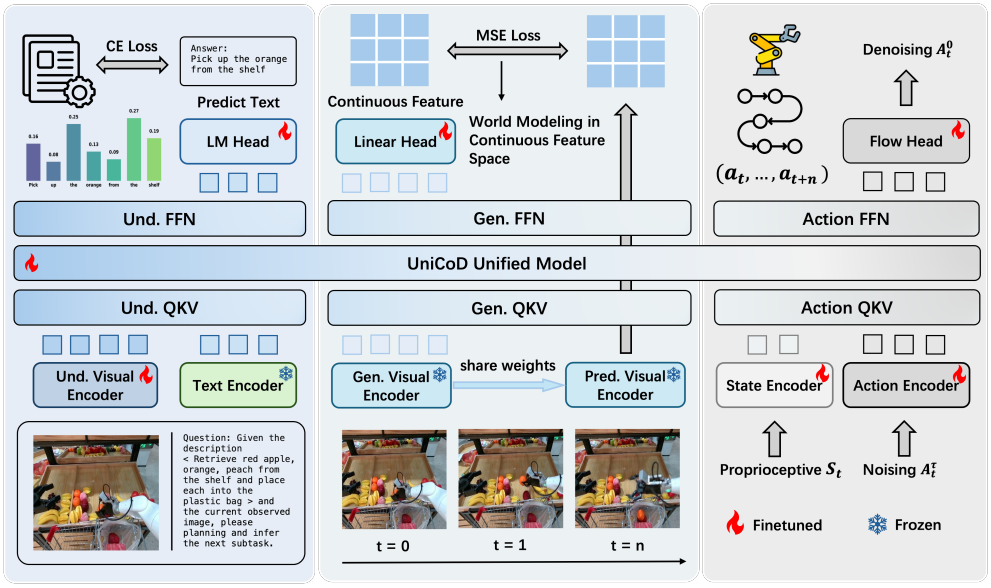

Figure 1: **Overview of the UniCode framework.**

In this section, we present the overall framework design and the two-stage training strategy of UniCoD, as illustrated in Figure xx. In the first stage, UniCoD is trained to learn joint text–image representations across diverse manipulation datasets. In the subsequent stage, an action expert is employed to integrate the multimodal inputs and predicted future states, generating the final robot actions.

3.1 UNIFIED VISION LANGUAGE JOINT EMBEDDING MODELING

Before introducing the robot action space, we first establish a cross-embodiment pre-training paradigm for robots. In this stage, a subset of the model parameters $U_{v,l}$ is jointly optimized through VQA and TI2E tasks. Formally, given a language instruction l and the current multi-view observations $o_t = \{o_1, o_2\}$ at time t, UniCoD is trained to predict the joint visual–text embedding:

$$\hat{o}_{t+h}, \hat{l} = U_{v,l}(o_t, l).$$

Here, $\hat{o}_{t+h} = V(o_{t+h}) = \{c_1, c_2, \ldots, c_n\}$ denotes the continuous representation encoded by the pre-trained visual encoder V, while $\hat{l} = \{d_1, d_2, \ldots, d_m\}$ corresponds to the m-token textual sequence.

Discrete Representation Learning. To endow the model with fundamental vision–language alignment capabilities, we initialize $U_{v,l}$ from the pre-trained vision-language model. The fine-grained language output comes partly from existing visual-language question-answer pairs and partly from planning and scene descriptions obtained from embodied tasks.

Worldmodeling under continous space.

Training Objective. The visual and language inputs are processed respectively through the mixture-of-transformers framework, then autoregresively generate $\hat{l}_{t+h}^{pred} = d_{1:m}^{pred}$, while the generation expert obtains the $\hat{o}_{t+h}^{pred} = c_{1:n}^{pred}$. We follow the standard setup of generative–understanding models,

employing cross-entropy loss for the language branch and mean squared error loss for the generative branch. This optimaze progress can be formulated as:

$$\mathcal{L}_1 = \lambda_1 \cdot \frac{1}{n}\sum_{i=1}^{n}\left\|c_i^{pred} - c_i\right\|_2^2 - (1-\lambda_1)\cdot\frac{1}{m}\sum_{j=1}^{m} logP_\theta(d_j \mid d_{<j}, l, o_t) \tag{1}$$

where λ_1 serves as a weighting factor to balance the loss contributions of the discrete and continuous representations.

3.2 Unified action modeling

In the previous stage, we obtained $U_{v,l}$ through pre-training, which endowed the model with basic capabilities in future state prediction and vision–language alignment. However, $U_{v,l}$ cannot yet be directly mapped to the action space. To address this limitation, in the second stage we fine-tune $U_{v,l}$ on embodiment data comprising visual, language, and action modalities, while simultaneously training an action expert from scratch to construct $U_{v,l,a}$.

Action & State Expert. Similar to the generation and understanding experts, we employ distinct attention weights to project actions and states (i.e., proprioception) into a shared attention space. Unlike the other experts, the action expert leverages flow matching to capture the continuous and inherently multi-modal distribution of the action space. Proprioceptive signals s_t are processed by an MLP-based state expert encoder, enabling fusion within the unified model. Given an action sequence $A_t = (a_t, a_{t+1}, \ldots, a_{t+h})$ to be executed, along with the observation o_t and instruction l, the unified model $U_{v,l,a}$ is trained to approximate vector fields as:

$$\mathcal{L}_{\text{flow}} = \mathbb{E}_{\tau\sim\mathcal{U}(0,1)}\,\mathbb{E}_{\{A_t,o_t,s_t,l\}\sim\mathcal{D}}\left[\left\|U_{v,l,a}(A_t^\tau, o_t, s_t, l, \tau) - (A_t - A_t^\tau)\right\|_2^2\right], \tag{2}$$

where $A_t^\tau = (1-\tau)\epsilon + \tau A_t$ denotes the interpolated actions at step τ, and $\epsilon \sim \mathcal{N}(0, I)$.

In this action training stage, we also jointly optimize the generation expert by predicting the future observation states $c_{1:n}$, yielding the following objective:

$$\mathcal{L}_2 = \lambda_2 \cdot \frac{1}{n}\sum_{i=1}^{n}\left\|c_i^{\text{pred}} - c_i\right\|_2^2 + (1-\lambda_2)\mathcal{L}_{\text{flow}}. \tag{3}$$

3.3 IMPLEMENTATION DETAILS

Model Setting. UniCoD employs Paligemma Beyer et al. (2024) as the VLM expert. For future observation encoding, we experiment with SigLIP Tschannen et al. (2025), DINOv3 Siméoni et al. (2025), and direct pixel-level prediction. Considering the information flow across modalities, we adopt a block-wise masking mechanism in the MoT attention: within each modality, bidirectional attention is applied, while across modalities a causal mask is enforced following the order of image, language, image prediction, state information, and action.

Training Data. In the pretraining stage, we utilize three categories of data to acquire joint text–image representations: (1) robot videos paired with fine-grained subtask descriptions and overall task instructions, which yield VQA and TI2E data for the generation–understanding task; (2) robot and human operation videos accompanied by task instructions, which are used as TI2E data; and (3) generic vision–language question answering data, employed for co-training to preserve the fundamental capabilities of the VLM. In the action modeling stage, we exclusively adopt VLA data collected in both simulation and real-world robotic environments. Further details regarding the datasets are provided in the supplementary material.

4 Experiment

To comprehensively evaluate our proposed method, XXX, we conduct extensive experiments across two simulation benchmarks and on two distinct real-world robotic platforms. Our experiments are designed to assess the performance of XXX and validate the effectiveness of our proposed modules.

4.1 Experimental Setup

Our experiments are conducted and deployed across four distinct environments. Figure 2 illustrates a selection of tasks from both our simulation and real-world settings.

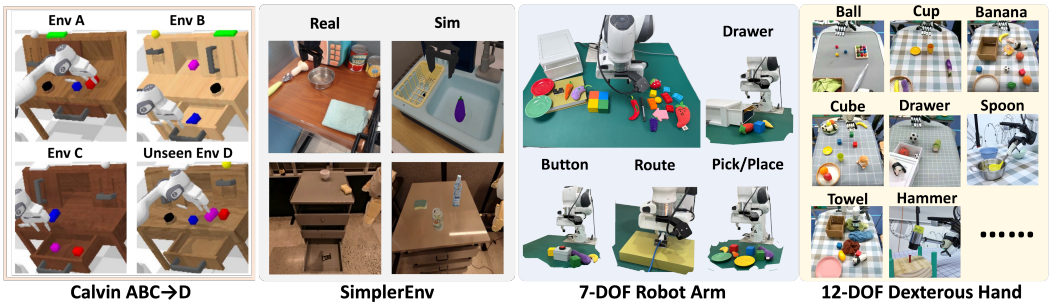

Figure 2: Our evaluation environments, including 2 simulation benchmarks and 2 real-world embodiments.

Calvin Benchmark Calvin is a simulation benchmark designed for evaluating long-horizon, language-conditioned manipulation policies. We employ the *ABC-D* split to evaluate the single-view generalization capabilities of the models. The evaluation suite includes 1,000 long-horizon sequences, each of length 5. We report the average length of completed sub-task sequences.

SimplerEnv Benchmark SimplerEnv is a simulation benchmark designed to evaluate policies trained on real-world datasets, such as Bridge-V2 and Fractal. The benchmark supports two types of robot arms: WindowX and Google Robot. For our evaluation, we conduct 240 runs for each task and report the average success rate.

Real-World Franka Emika Panda Arm We deploy models on a Franka Emika arm for real-world task comparison. We first collected a dataset of 2,000 trajectories spanning over 20 distinct tasks, encompassing six fundamental skills: picking, placing, opening a drawer, closing a drawer, pressing a button, and routing a cable. We evaluate performance on both seen and unseen task variations. The unseen category primarily involves grasping novel objects not present in the training data and introducing misleading objects. More details can be found in Appendix A.3.1.

Real-World XArm with 12-DOF X-Hand On our dexterous manipulation platform, we train different models using a dataset of 4,000 trajectories across more than 100 tasks. The models are then evaluated in a variety of seen and unseen scenarios, which cover 13 distinct skills in 9 categories. More details can be found in Appendix A.3.2.

4.2 Simulation Experiments

Implementation Details We first pre-train XXX following the methodology described in Section 3. Subsequently, we fine-tune the model on 8 A100 GPUs for 22k steps, using a learning rate of 5×10^{-5} and a batch size of 1024. For all simulation training, we consistently use a single, third-person-view image of size 224×224 as the visual input. In Calvin, we use an action chunk size of 10, and during deployment, the full 10-step chunk is executed at each inference step. In SimplerEnv, we use an action chunk size of 4; for the WindowX environment (corresponding to the Bridge dataset), the full 4-step chunk is executed, whereas for the Google Robot environment (corresponding to the Fractal dataset), half of the action chunk is executed.

Baselines We compare XXX against several state-of-the-art VLAs and prediction-based policies. On SimplerEnv, we benchmark XXX against RT-1-X (Brohan et al., 2022), Octo (Team et al., 2024), OpenVLA (Kim et al., 2024), RoboVLMs (Liu et al., 2025), SpatialVLA (Qu et al., 2025), π_0 (Black et al., 2024), CogAct (Li et al., 2024) and Villa-x (Chen et al., 2025). On Calvin, we compare XXX against several policies that leverage visual generation tasks, including GR-1 (Wu et al., 2023), π_0 (Black et al., 2024), VPP (Hu et al., 2024), and UP-VLA (Zhang et al., 2025). To ensure a fair comparison, we reproduce these baselines and standardize their visual input to a single third-person

view. For π_0, we specifically use the implementation from the open-pi-zero and report its performance under the same training and evaluation setup used in XXX for a direct comparison.

4.2.1 PERFORMANCE ON SIMULATION BENCHMARKS

Model	Carrot on Plate		Eggplant in Basket		Spoon on Towel		Stack Cube		Success
	Grasp	**Success**	**Grasp**	**Success**	**Grasp**	**Success**	**Grasp**	**Success**	**Average**
RT-1-X	20.8	4.2	0.0	0.0	16.7	0.0	8.3	0.0	1.1
Octo-Base	52.8	8.3	66.7	43.1	34.7	12.5	31.9	0.0	16.0
OpenVLA	33.3	0.0	8.3	4.1	4.1	0.0	12.5	0.0	1.0
RoboVLMs	33.3	20.8	91.7	79.2	70.8	45.8	54.2	4.2	37.5
SpatialVLA	29.2	25.0	100.0	**100.0**	20.8	16.7	62.5	29.2	42.7
π_0*	58.5	48.8	78.8	64.6	83.3	73.3	62.5	12.5	49.8
CogAct	/	58.3	/	45.8	/	29.2	/	**95.8**	57.3
Villa-x	/	46.3	/	64.6	/	77.9	/	61.3	62.5
XXX (Ours)	75.0	**63.0**	100.0	89.6	83.3	**78.8**	91.7	52.5	**71.0**

Table 1: Results on SimplerEnv-WindowsX (visual matching). Entries marked with * are methods reproduced with our training and test settings.

Tables 1 and 2 present the performance of our method on the SimplerEnv-WindowX and SimplerEnv-Google Robot benchmarks, respectively. We report the officially published results of other methods for comparison. On both robotic platforms, our method achieves the highest success rates of 71.0% and 78.4%, attaining state-of-the-art (SOTA) performance. We highlight the top-performing and second-best methods for each task category in **bold** and with an underline. It is evident that XXX demonstrates consistently high success rates across all sub-tasks. This contrasts with other methods, which often exhibit *"spiky"* performance profiles—excelling on some tasks while performing poorly on others. This finding underscores the superior multi-task learning capabilities of our approach.

Furthermore, for a fair, apple-to-apple comparison with the architecturally similar π_0 baseline, we reproduced it within our identical training and evaluation framework. Across both environments, we found that the novel components in XXX yield a significant performance uplift of over 20%. We also observed that this improvement is consistently present at every training checkpoint, indicating that the stable gains can be attributed to our method's ability to learn continuous future features and discrete representations simultaneously.

We also compare XXX against several policies that leverage advanced vision-based training methodologies on the Calvin ABC-D split, with results shown in Table 3. Since many prior works utilize multi-view images and historical information, we re-implemented these baselines using a standardized single, third-person-view image as visual input to ensure a fair comparison of the benefits conferred by our training method. The results demonstrate that XXX achieves the best performance on single-view manipulation tasks within the Calvin benchmark. Moreover, when compared to the baseline `pi0`, our method again exhibits a performance improvement, consistent with the results on SimplerEnv.

Model	Pick Coke	Move Near	O./C. Drawer	Put in Drawer	AVG↑
RT-1-X	56.7	31.7	59.7	21.3	42.4
Octo-Base	17.0	4.2	22.7	0.0	11.0
OpenVLA	16.3	46.2	35.6	0.0	24.5
RoboVLMs	77.3	61.7	43.5	24.1	51.7
π_0*	93.3	78.1	23.6	12.5	51.9
CogACT	91.3	**85.0**	**71.8**	50.9	74.8
Villa-x	**98.7**	75.0	59.3	5.6	59.6
XXX (Ours)	**98.7**	81.5	63.2	**70.0**	**78.4**

Table 2: Results on SimplerEnv-Google Robot (visual matching). Entries marked with * are methods reproduced with our training and test settings.

4.3 REAL WORLD EXPERIMENTS

Implementation Details We fine-tune the pre-trained XXX model separately on the datasets collected from our two real-world robotic platforms to evaluate its performance on a variety of seen and unseen tasks. The fine-tuning process is conducted for

Method	Tasks completed in a row					Avg. Len ↑
	1	**2**	**3**	**4**	**5**	
RT-1*	0.533	0.222	0.094	0.038	0.013	0.90
DP*	0.402	0.123	0.026	0.008	0.000	0.56
GR-1	0.854	0.712	0.596	0.497	0.401	3.06
π_0*						
VPP*						
UP-VLA*	0.928	0.865	0.815	0.769	0.699	4.08
XXX (Ours)						

Table 3: Long-horizon evaluation on the Calvin ABC→D benchmark. Entries marked with * are methods reproduced with our training and test settings. We take use of a single

10 epochs using a batch size of 1024 and a learning rate of 5×10^{-5}, with both the prediction horizon and action chunk length set to 10. For the Franka Emika Panda arm, the model is fine-tuned on 2,000 trajectories, and during deployment, we evaluate both full and half action chunk execution, reporting the superior result. On the XArm with a 12-DOF dexterous hand, we use a larger dataset of 4,000 trajectories and execute the full 10-step action chunk at each inference step. We test on seen tasks, which involve familiar objects in novel, randomized positions, and unseen tasks, which introduce novel color, objects, and background. For each task configuration, we conduct 20 trials from randomized initial configurations and report the average task success rate. More details can be found in Appendix A.3.

4.3.1 Performance on Real World Experiments

We compare `XXX` against `Diffusion Policy (DP)` (Chi et al., 2023), `GR-1` (Wu et al., 2023), π_0 (Black et al., 2024), and `VPP` (Hu et al., 2024) in two environments, visualizing the results in Figure 3 and 4. Our method achieves the highest overall task success rates on both real-world

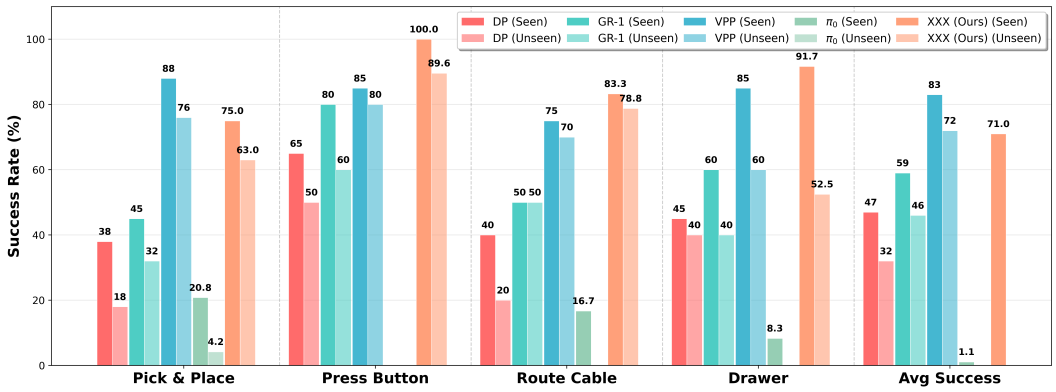

Figure 3: Results on real-world 7DOF robotarm experiment. More detailed quantitative results are provided in Table 6.

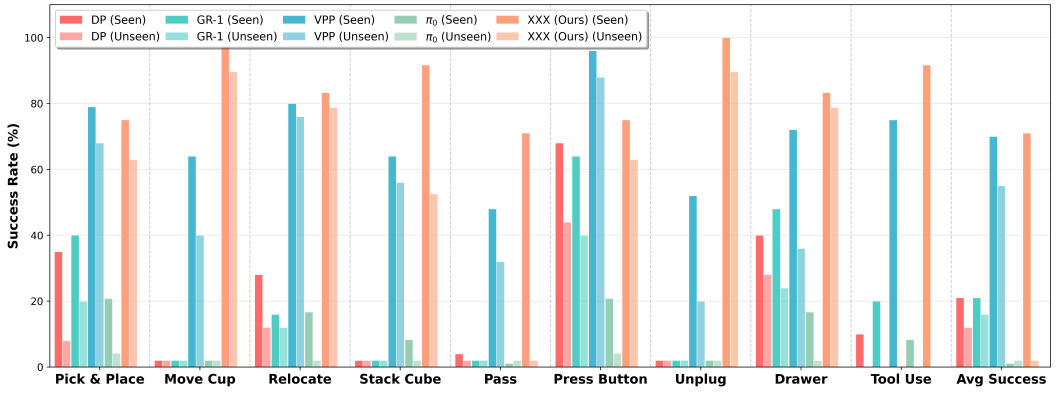

Figure 4: Results on real-world 12-DOF dexterous hands experiment. More detailed quantitative results can be found in Table 7.

robotic platforms. Specifically, on the Franka Panda arm, `XXX` attains the best performance across all four task categories, outperforming baselines on both seen and unseen tasks. This demonstrates that our approach effectively enhances both multi-task learning and generalization capabilities.

Consistent with our findings in the Simpler simulation environment, our method again shows superior performance over the architecturally similar π_0 baseline across a majority of these real-world tasks. Furthermore, on the more complex 12-DoF dexterous hand platform, XXX achieves the highest average success rate across all nine skill categories. These consistent, state-of-the-art results across two morphologically distinct robots validate the effectiveness and broad applicability of our proposed method.

4.4 Ablation Study

In this section, we conduct a series of ablation studies to validate the effectiveness of the different components within XXX. These experiments investigate the role of our continuous visual representations, the impact of our large-scale pre-training phase involving both language and visual prediction, and a comparison of several continuous vision encoding methods proposed in Sec 3. All ablation studies are conducted in the Simpler simulation environment, following the same training and evaluation protocols described in Sec 4.2.

Model	Carrot	Eggplant	Spoon	Cube	AVG↑
w/o Pretrain					
w/o Continuous	48.8	64.6	73.3	12.5	49.8
w/o Continuous w/ Pred	52.5	79.2	79.6	30.0	60.3
XXX	60.8	87.1	78.8	50.4	69.3
w/ Pretrain					
XXX (Ours)	**63.0**	**89.6**	**78.8**	**52.5**	**71.0**

Table 4: Ablation study on unified pretraining paradigm and continuous feature for prediction.

Effectiveness of Continuous Predictive Visual Representations To validate the effectiveness of prediction using continuous representations, we compare a version of XXX without pre-training against two baselines, as shown in Table 4. We evaluate the following without using pretraining: (1) `w/o Continuous` (π_0), where the modules for predicting continuous future features (including the auxiliary prediction expert and its corresponding encoder/decoder) are removed. (2) `w/Pred`, which predicts future raw pixels using a two-layer MLP. This helps us elucidate the trade-offs between using high-level visual features versus raw pixels as the predictive signal. The results in `w/o Pretrain` section of the table show that our proposed continuous visual feature prediction boosts performance by approximately 20%. Furthermore, the comparison with `w/Pred` reveals that continuous features are indeed a more effective signal for future prediction, enabling the model to extract dynamic information crucial for action generation.

Effectiveness of Large-Scale Planning and Prediction Pre-training Table 4 also presents a comparison between XXX with and without pre-training. Overall, pre-training improves the success rate across all tasks, yielding a performance gain of approximately 2%. During fine-tuning, we observe that leveraging large-scale external data for future and language prediction accelerates the model's convergence on the robotics dataset. This effect is particularly pronounced in the convergence of the future prediction loss. This indicates that our joint pre-training scheme, which combines continuous and discrete prediction, provides a superior model initialization, especially for the prediction expert module, which translates to tangible benefits during downstream fine-tuning.

Method	Google robot					WidowX robot				
	Pick	Move	Drawer	Put	**AVG**	Carrot	Eggplant	Spoon	Cube	**AVG**
XXX-Distill	97.2	82.6	**61.9**	**74.4**	**79.0**	48.8	**95.8**	**89.6**	34.6	67.2
XXX-Dino	**98.3**	**80.2**	51.1	63.3	73.2	54.6	81.7	78.8	49.6	66.1
XXX-Siglip	97.7	80.2	61.3	72.4	77.9	**60.8**	87.1	78.8	**50.4**	**69.3**

Table 5: Ablation study on choice of continuous vision features.

Choice of Continuous Visual Prediction We further compare the different encoding methods for future prediction proposed in our methodology. Specifically, we evaluate three distinct approaches (all without pre-training), with results on both Simpler environments shown in Table 5: (1) `XXX-Distill`, which takes the input embeddings of the ViT (from the current frame) as input to the prediction expert and predicts the output features of ViT for the future frame. This approach is analogous to distilling knowledge from the ViT encoder itself. (2) `XXX-Dino` and (3) `XXX-Siglip`,

which take the output features of their respective vision encoders (DINO Siméoni et al. (2025) or SigLIP Tschannen et al. (2025)) for the current frame as input to predict the corresponding features for the future frame.

The results show that `XXX-Siglip` demonstrates better performance on both benchmarks, and consequently, we select SigLIP as the vision encoder for our `XXX` model. Notably, on Google Robot environment, `XXX-Distill` achieves better performance than the `XXX-Siglip` when neither is pre-trained. This suggests that the distillation-style architecture has inherent advantages. In contrast, `XXX-Dino` performs significantly worse than the other two. This is likely because the DINO feature space is not aligned with the VLM backbone. Conversely, since SigLIP is the native vision encoder for Paligemma, its feature space is naturally more aligned with that of the VLM expert, facilitating more effective integration within the prediction expert.

5 Conclusion

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments, including those to funding agencies, go at the end of the paper.

References

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

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.

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.

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.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

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.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

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.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36, 2024.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in Neural Information Processing Systems*, 36, 2024.

Yanjiang Guo, Yucheng Hu, Jianke Zhang, Yen-Jen Wang, Xiaoyu Chen, Chaochao Lu, and Jianyu Chen. Prediction with action: Visual policy learning via joint denoising process. *arXiv preprint arXiv:2411.18179*, 2024.

Yanjiang Guo, Jianke Zhang, Xiaoyu Chen, Xiang Ji, Yen-Jen Wang, Yucheng Hu, and Jianyu Chen. Improving vision-language-action model with online reinforcement learning. *arXiv preprint arXiv:2501.16664*, 2025.

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.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Peiyan Li, Yixiang Chen, Hongtao Wu, Xiao Ma, Xiangnan Wu, Yan Huang, Liang Wang, Tao Kong, and Tieniu Tan. Bridgevla: Input-output alignment for efficient 3d manipulation learning with vision-language models. *arXiv preprint arXiv:2506.07961*, 2025.

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.

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.

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. *arXiv preprint arXiv:2412.14058*, 2025.

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.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

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.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

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*, 2025a.

Yuqi Wang, Xinghang Li, Wenxuan Wang, Junbo Zhang, Yingyan Li, Yuntao Chen, Xinlong Wang, and Zhaoxiang Zhang. Unified vision-language-action model. *arXiv preprint arXiv:2506.19850*, 2025b.

Junjie Wen, Yichen Zhu, Jinming Li, Zhibin Tang, Chaomin Shen, and Feifei Feng. Dexvla: Vision-language model with plug-in diffusion expert for general robot control. *arXiv preprint arXiv:2502.05855*, 2025.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. *arXiv preprint arXiv:2312.13139*, 2023.

Jianke Zhang, Yanjiang Guo, Xiaoyu Chen, Yen-Jen Wang, Yucheng Hu, Chengming Shi, and Jianyu Chen. Hirt: Enhancing robotic control with hierarchical robot transformers. *arXiv preprint arXiv:2410.05273*, 2024.

Jianke Zhang, Yanjiang Guo, Yucheng Hu, Xiaoyu Chen, Xiang Zhu, and Jianyu Chen. Up-vla: A unified understanding and prediction model for embodied agent. *arXiv preprint arXiv:2501.18867*, 2025.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 1702–1713, 2025.

Zhongyi Zhou, Yichen Zhu, Minjie Zhu, Junjie Wen, Ning Liu, Zhiyuan Xu, Weibin Meng, Ran Cheng, Yaxin Peng, Chaomin Shen, et al. Chatvla: Unified multimodal understanding and robot control with vision-language-action model. *arXiv preprint arXiv:2502.14420*, 2025.

A Appendix

A.1 Qualitative Comparison of Encoded Future Visual Representations

To qualitatively analyze the characteristics of different encoding methods, we visualize the features they produce. Specifically, we compare features from a single robot trajectory encoded in three ways: raw image pixels, continuous visual features from a ViT encoder, and discrete visual tokens from a VQ-GAN. We selected a trajectory from the `Fractal` dataset corresponding to the instruction *pick the coffee bag from the drawer onto the table*. For each frame, the resulting features—raw pixels (flattened from $224 \times 224 \times 3$), ViT features (flattened from 256×1152), and VQ-VAE tokens (2048-dim)—are first reduced to 50 dimensions via PCA and then projected into a 2D space using t-SNE for visualization.

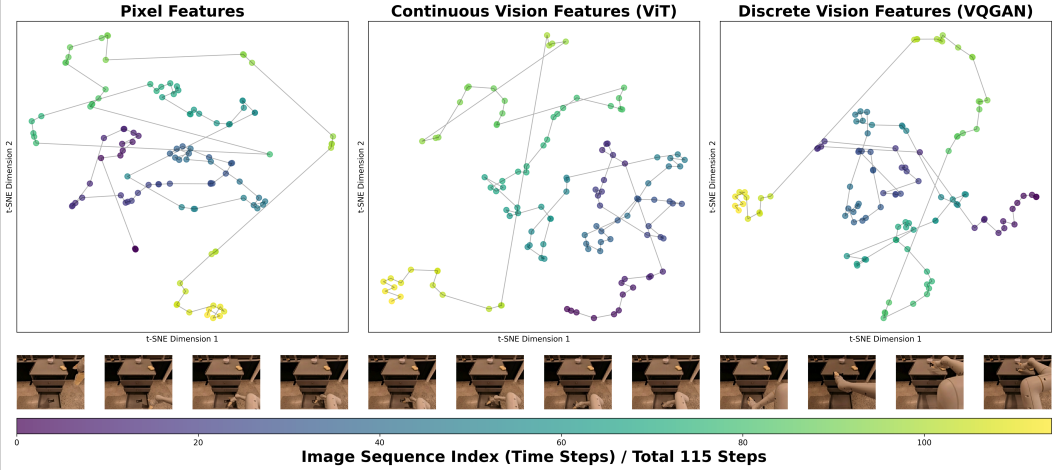

Figure 5: t-SNE Visualization of Different Future Representations.

Figure 5 illustrates the t-SNE visualizations for the trajectory encoded by these three methods. To highlight the temporal evolution, feature points from adjacent frames are connected by lines.

- **Pixel Features (Left):** This encoding preserves the most low-level information. We observe that despite small visual changes between consecutive frames, the corresponding pixel-level features exhibit high variance, often jumping into regions occupied by features from distant timesteps. This suggests that using raw pixel values as a predictive signal could mislead the policy by causing it to over-emphasize low-level, high-frequency changes.
- **ViT vs. VQ-GAN Features (Center and Right):** A comparison reveals a distinct *"circling phenomenon"* in the VQ-GAN visualization, where features from many different timesteps collapse into a dense central region. This indicates poor temporal separability in context of manipulation trajectories. In contrast, the ViT features provide the best separation of the three methods, organizing features from different frames into distinct, minimally-overlapping clusters.

This qualitative analysis supports our insight that continuous features, by virtue of focusing on high-level semantic information, serve as a more stable and suitable predictive signal for robot action policies within our framework.

A.2 Details about Simulation Benchmarks

Calvin Benchmark Calvin is a simulation benchmark designed for evaluating long-horizon, language-conditioned manipulation policies. It comprises four distinct environments (A, B, C, and D) and offers evaluation splits such as *ABC-D* and *ABCD-D*. In our experiments, we employ the *ABC-D* split to evaluate the single-view generalization capabilities of the models. Models are trained on data collected from environments A, B, and C, and subsequently evaluated in the unseen

environment D. This evaluation suite includes 34 different manipulation tasks organized into 1,000 long-horizon sequences, each of length 5. We report the average length of successfully completed sub-task sequences.

SimplerEnv Benchmark SimplerEnv is a simulation benchmark designed to evaluate policies trained on large-scale real-world datasets, such as Bridge-V2 and Fractal. It procedurally generates scenes that mimic real-world environments using texturing techniques, allowing models trained on real data to be tested directly in simulation without requiring physical deployment. The benchmark supports two types of robot arms: the WindowX and the Google Robot. For our evaluation, we conduct 240 runs for each task and report the average success rate.

A.3 Details on Real World Experiments

A.3.1 Franka Panda Robot Arm

Real-World Franka Emika Panda Arm We deploy several models on a Franka Emika Panda arm for real-world task comparison. The robot arm features 7 degrees of freedom (DoF). Its action space is defined by a 7-dimensional vector, where the first six dimensions specify the relative change in the end-effector's 6D pose (3D position and 3D orientation), and the final dimension controls the binary state of the gripper (open or closed). In our experiments, the policy takes images from an on-board, first-person-view camera as visual input and outputs these relative actions. We first collected a dataset of 2,000 trajectories spanning over 20 distinct tasks, encompassing six fundamental skills: picking, placing, opening a drawer, closing a drawer, pressing a button, and routing a cable. We evaluate performance on both seen and unseen task variations. The unseen category primarily involves grasping novel objects not present in the training data.

The task suite for the Franka Panda arm includes:

- **Pick & Place:** Grasping and placing a variety of objects. The training set includes items such as a toy banana, a toy eggplant, red/green/blue blocks, and red/yellow/black plates.
- **Press Button:** Pressing a toy button using a grasped black block as a tool.
- **Route Cable:** Routing a thin black rubber cable into a narrow slot.
- **Drawer Operation:** Opening a toy drawer.

Unseen Tasks These are designed to evaluate generalization: *Novel Objects:* Grasping objects not seen during training (e.g., toy chili, toy strawberry, yellow block, large toy eggplant, arrow sticker, marker pen). *Distractors:* Operating in the presence of irrelevant distractor objects. *Visual Variations:* Adapting to changes in background color and object color.

We tested `XXX`, `Diffusion Policy (DP)` (Chi et al., 2023), `GR-1` (Wu et al., 2023), π_0 (Black et al., 2024), and `VPP` (Hu et al., 2024) on this environment. The detailed results are shown in Table 6 (corresponding to Figure 3).

Model	Pick & Place		Press Button		Route Cable		Drawer		**Avg Success**	
	Seen	Unseen	Seen	Unseen	Seen	Unseen	Seen	Unseen	**Seen**	**Unseen**
DP	38	18	65	50	40	20	45	40	47	32
GR-1	45	32	80	60	50	50	60	40	59	46
VPP	88	76	85	80	75	70	85	60	83	72
π_0	20.8	4.2	0.0	0.0	16.7	0.0	8.3	0.0	1.1	0
XXX (Ours)	75.0	63.0	100.0	89.6	83.3	78.8	91.7	52.5	71.0	0

Table 6: Detailed results on Franka-Emika Panda Robotarm. We evaluate each task 20 times (100 trials per skill) with random initialization and report the average success rate.

A.3.2 XArm Dexterous Manipulation

Real-World XArm with 12-DOF X-Hand Our 12-DoF single-arm dexterous manipulation platform, which comprises a 7-DoF XArm and a 5-DoF hand, is controlled using a dual-view visual

input from both first-person and third-person cameras. During evaluation, we test pick-and-place capabilities across 5 distinct task variations for a total of 50 trials. For all other skills, we conduct 20 trials per task. The final performance is reported as the average success rate for each skill. We train different models using a dataset of 4,000 trajectories across more than 100 tasks. The models are then evaluated in a variety of seen and unseen scenarios, which cover 13 distinct skills, e.g., picking, placing, stacking, and pouring. To specifically test for visual generalization, we alter the background colors and novel objects during evaluation in the unseen scenarios.

The task suite for the XArm platform includes:

- **Dexterous Pick & Place:** Dexterously grasping and placing a wide range of objects. The training set includes a toy banana, a toy eggplant, a toy orange, small and large toy soccer balls, a computer mouse, a toy drawer, and more.
- **Move Cup:** Grasping and moving a cup to a different location.
- **Relocate:** Grasping an object and placing it adjacent to another target object.
- **Stack Cube:** Placing one block on top of another.
- **Pass:** Grasping an object and handing it to a human operator.
- **Press Button:** Directly actuating a toy button with a finger.
- **Unplug:** Extracting a rubber cable from a socket.
- **Drawer Operation:** Opening or closing a toy drawer.
- **Tool Use:** Using various tools, such as a spoon (e.g., for scooping) and a toy hammer (e.g., for striking).

Model	Pick & Place		Move Cup		Relocate		Stack Cube		Pass	
	Seen	Unseen	Seen	Unseen	Seen	Unseen	Seen	Unseen	Seen	Unseen
DP	35	8	0	0	28	12	0	0	4	0
GR-1	40	20	0	0	16	12	0	0	0	0
VPP	79	68	64	40	80	76	64	56	48	32
π_0	20.8	4.2	0.0	0.0	16.7	0.0	8.3	0.0	1.1	0
XXX (Ours)	75.0	63.0	100.0	89.6	83.3	78.8	91.7	52.5	71.0	0

Model	Press Button		Unplug		Drawer		Tool Use		**Avg Success**	
	Seen	Unseen	Seen	Unseen	Seen	Unseen	Seen	Unseen	**Seen**	**Unseen**
DP	68	44	0	0	40	28	10	/	21	12
GR-1	64	40	0	0	48	24	20	/	21	16
VPP	96	88	52	20	72	56	75	/	70	55
π_0	20.8	4.2	0.0	0.0	16.7	0.0	8.3	/	1.1	0
XXX (Ours)	75.0	63.0	100.0	89.6	83.3	78.8	91.7	/	71.0	0

Table 7: Detailed results on XArm with dexterous hand. We evaluate 50 times on Pick & Place tasks and 20 trials on other tasks with random initialization and report the average success rate.