PRIGO: PRIMITIVE GUIDANCE TO DIFFUSION AND FLOW POLICIES FOR ROBOTIC MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Imitation learning has enabled remarkable progress in robotic manipulation, especially with diffusion- and flow-based policies that generate complex visuomotor behaviors directly from demonstrations. Despite their strong performance, these policies often struggle to generalize across tasks and environments. A key aspect that remains underexplored is the local motion structure inherent in manipulation behaviors, which is not explicitly captured by continuous action prediction. Inspired by the compositional structure of human behaviors, we propose **PriGo**, a primitive-guided test-time adaptive framework for robust robotic manipulation. PriGo introduces PANet, a lightweight primitive prediction module that infers primitive distributions directly from observations. We further propose a differentiable primitive guidance mechanism that refines generated actions during inference, steering trajectories toward semantically consistent behaviors. Unlike prior primitive-conditioned approaches, PriGo operates exclusively at test time and can be seamlessly integrated with pretrained diffusion- and flow-based policies without retraining the backbone policies. Extensive experiments on LIBERO, CALVIN, SIMPLER, and real-world robotic tasks demonstrate that PriGo consistently improves robustness and generalization across both diffusion- and flow-based policies. Codes are available on **PriGo**.

1 INTRODUCTION

Imitation learning (IL) has become a dominant paradigm for robotic manipulation, enabling policies to acquire complex visuomotor behaviors directly from expert demonstrations without manually designed reward functions. Recent diffusion-based and flow matching-based policies have further improved action generation quality by modeling multi-modal action distributions and leveraging large-scale vision-language representations. Despite these advances, existing policies often remain brittle under distribution shifts, including variations in objects, lighting, distractors, and unseen task (Sapkota et al., 2025; Li et al., 2025b).

One possible limitation is that continuous action prediction does not explicitly represent the local motion structure underlying manipulation behaviors. Consequently, generated actions may locally resemble expert demonstrations while failing to preserve the intended motion structure required for successful task completion. As shown in Figure 1.a, when opening a door, the policy incorrectly pulls before rotating the key, or generates unstable grasping motions under environmental perturbations. Such failures become particularly severe in long-horizon manipulation, where small local errors accumulate across multiple subtasks. To improve robustness and generalization, an important direction is to incorporate structured action priors into policy generation. Human manipulation behaviors naturally exhibit compositional structure, where complex tasks are composed of a small number of reusable motion primitives such as grasping, pushing, pulling, and rotation. Rather than treating actions as unconstrained continuous trajectories, these primitives provide a compact structural prior that constrains policies toward semantically consistent behaviors.

Recent works Hiranaka et al. (2023); Gao et al. (2024) have sought to disentangle primitives into discrete types with associated parameters, capturing both categorical abstractions and action prediction. However, they limit the primitive set to basic actions, which restricts their applicability. Furthermore, these methods treat primitives as additional input rather than enforcing them as mandatory constraints, resulting in generated actions that do not adhere to the required primitives.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To this end, we introduce PriGo, a primitive-guided test-time adaptation framework for robotic manipulation policies. Instead of retraining existing policies, PriGo introduces structured primitive guidance during inference in a plug-and-play manner. Specifically, we first train a lightweight primitive classifier, PANet, which predicts primitive distributions directly from visual observations and language instructions. We adopt a compact primitive taxonomy consisting of eight manipulation primitives, providing a balance between structural expressiveness and cross-task generalization.

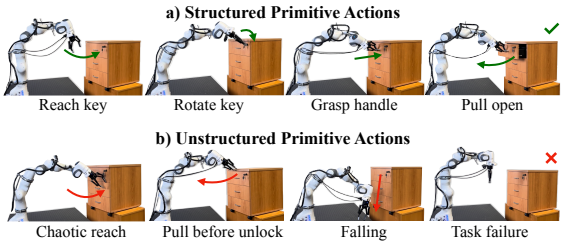


Figure 1: Comparison of a) our policy with PriGo, which produces structured primitive actions, and b) vanilla policies without PriGo, which lack structured primitive action outputs.

We then incorporate differentiable primitive guidance into both diffusion and flow policies. During inference, actions are refined using gradients derived from primitive consistency objectives, encouraging trajectories to align with the predicted manipulation structure. Unlike prior guided diffusion approaches that mainly rely on task rewards or language supervision, PriGo introduces structured action-level guidance. Our contributions are summarized as follows:

- We propose PriGo, a plug-and-play primitive-guided adaptation framework that bridges discrete manipulation structures with continuous generative action refinement.
- We introduce a differentiable primitive guidance mechanism that constrains diffusion and flow policy generation toward structured manipulation behaviors without retraining policies.
- We design a lightweight primitive classification framework that predicts probabilistic primitive distributions and supports multi-primitive guidance.
- We conduct extensive evaluations across multiple simulation benchmarks and real-robot tasks, demonstrating consistent improvements over the corresponding pretrained backbone policies.

2 Related Works

Diffusion and Flow Policies for Manipulation. Diffusion policies (DPs) Chi et al. (2023); Black et al. (2024b); Xian et al. (2023); Wang et al. (2025); Ze et al. (2024); Yang et al. (2024); Ke et al. (2024); Wang et al. (2024); He et al. (2025); Xiang et al. (2026); Team et al. (2024); Wen et al. (2024); Li et al. (2024); Zhou et al. (2025) have recently emerged as an effective framework for visuomotor policy learning by modeling action generation as iterative denoising processes. Existing approaches can be broadly divided into two categories. The first is `lightweight DPs` Chi et al. (2023); Black et al. (2024b); Xian et al. (2023); Wang et al. (2025); Ze et al. (2024); Yang et al. (2024); Ke et al. (2024); Wang et al. (2024); He et al. (2025); Xiang et al. (2026); Lv et al. (2026), which have evolved from the original 2D diffusion policy to 3D variants and, more recently, to equivariant formulations that better exploit symmetries. The second category comprises `Vision-Language-based DPs` Team et al. (2024); Wen et al. (2024); Li et al. (2024); Zhou et al. (2025); Liu et al. (2024; 2025a); Gu et al. (2026), which combine diffusion with large-scale vision–language models (VLM) to enable general-purpose robotic behaviors across diverse tasks.

Alongside diffusion-based approaches, flow matching has gained increasing popularity for robotic policy learning, owing to its stable optimization and efficient inference. Existing approaches can be broadly categorized into two groups: (1) lightweight flow-based policies, such as FlowMS Rouxel et al. (2024), FMP Zhang & Gienger (2024), and ActionFlow Funk et al. (2024), which focus on stable trajectory generation; and (2) VLM-integrated flow policies and VLA systems, including π series of works Black et al. (2024a; 2025), HiRobot Shi et al. (2025), GraspVLA Deng et al. (2025), SmolVLA Shukor et al. (2025), GR00T Bjorck et al. (2025), and LingBot-VA Li et al. (2026), which have established flow matching combined with large-scale vision-language representations as a leading paradigm for improving generalization, real-world deployment, and computational efficiency.

Despite their success, existing diffusion and flow policies remain sensitive to distribution shifts at test time. Several recent works attempt to address this issue through guided sampling or test-time adaptation. MetaDiffuser Ni et al. (2023) employs classifier guidance, using pretrained models to steer the diffusion process, while LDuS Kim et al. (2024) leverages large language models (LLMs)

to bias sampling toward goal-consistent actions. More recently, ADPro Li et al. (2025c) proposes task-aware initialization and constrains updates, yielding a more generalizable policy. However, these approaches primarily focus on task-level objectives or reward guidance, and do not explicitly enforce structured manipulation behaviors during action generation.

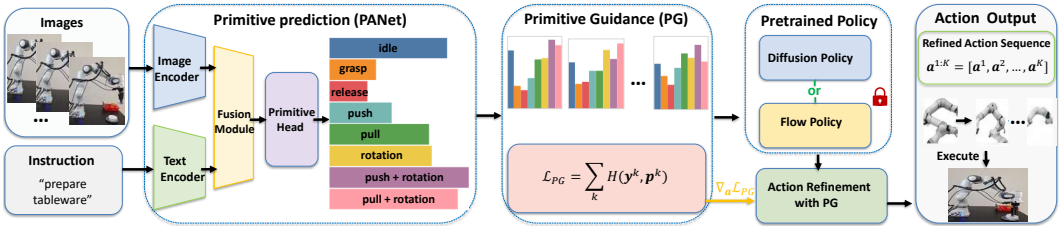


Figure 2: Overview of **PriGo**, a primitive-guided framework for robotic manipulation policies. Given visual observations and language instructions, the primitive classifier PANet predicts probabilistic manipulation primitives. During inference, generated actions from pretrained diffusion or flow policies are converted into soft primitive distributions and aligned with the predicted primitives through differentiable primitive guidance. The resulting gradient-based refinement steers action toward semantically consistent trajectories without retraining the backbone policy.

Primitive-based Policies. Primitive-based control has long been studied for improving policy compositionality and generalization. Early approaches Peng et al. (2019); Merel et al. (2019); Knaust & Koert (2021); Dalal et al. (2021) employed compositional networks to combine primitives into complex behaviors, where the composite policy's action distribution was a weighted sum over a set of **latent feature primitives**. However, their expressiveness was constrained by the simplicity of Gaussian primitives. Other researchers leverage LLMs to decompose task instructions into **prompt primitives**, using chain-of-thought reasoning to guide policy execution Zhang et al. (2023); Garcia et al. (2025); Pan et al. (2025), or use a lightweight router network to select skills Gu et al. (2026). While effective in some cases, they rely heavily on the reliability of LLM-based decomposition.

Several recent works further explore **extensible network primitives** that expand through incrementally added policy modules, often implemented with lightweight adapters such as LoRA Sun et al. (2025); Liu et al. (2025b); Zhou et al. (2025), enabling continual learning across diverse tasks Wan et al. (2024); Guo et al. (2025). Nevertheless, it raises challenges in storage efficiency and in managing experts' allocation as the library scales. More recent works Hiranaka et al. (2023); Gao et al. (2024) sought to disentangle primitives into discrete types with associated parameters. Yet, they typically restrict the primitive set to only a handful of basic actions.

Different from prior works, our work focuses on test-time primitive guidance rather than primitive-conditioned policy learning. PriGo does not require retraining the policy and instead introduces differentiable primitive constraints during inference. Furthermore, our formulation naturally supports multi-primitive guidance, enabling smooth transitions between manipulation primitives.

3 METHOD

As illustrated in Figure 2, PriGo introduces primitive-guided test-time adaptation for robotic manipulation policies. Our framework consists of two components: a primitive classifier PANet that predicts primitive distributions from observations, and a differentiable primitive guidance mechanism that constrains action generation during inference. PriGo can be seamlessly integrated into pretrained diffusion and flow policies without retraining the backbone policies.

Problem Statement. Given a set of expert demonstrations consisting of observation-action pairs, denoted as $\mathcal{D} = \{(\mathbf{O}^k, \mathbf{A}^k)\}_{k=1}^N$, our objective is to learn a manipulation policy that maps an observation $\mathbf{O}$ to an action $\mathbf{A}$. Here, $\mathbf{O}$ consists of RGB images and a language instruction, and $\mathbf{a} := (\mathbf{x}, \mathbf{r}, w)$ denotes a 7-DoF action, where $\mathbf{x}$ and $\mathbf{r}$ represent translational and rotational action increments, respectively, and $w \in [0, 1]$ denotes the normalized gripper command, with larger values corresponding to a more open gripper. To distinguish the temporal indices of robot execution from those of the generative process, we use the superscript k in $\mathbf{a}^k$ to denote the robot execution step, while the subscript t in $\mathbf{a}_t$ denotes the denoising or flow integration step.

3.1 PRIMITIVE REPRESENTATION AND CLASSIFICATION

Instead of directly constraining continuous actions, PriGo introduces a compact set of manipulation primitives as structured action priors $\mathcal{Y} = \{$"idle", "grasp", "release", "push", "pull", "rotation", "push+rotation", "pull+rotation"$\}$. We adopt this compact primitive taxonomy to balance structural expressiveness and cross-task generalization. Empirically, overly fine-grained primitive decompositions introduce long-tail imbalance and semantic ambiguity, while the proposed primitive set yields stable primitive distributions across diverse tasks.

Definition 1. ***(Primitive Action Hard Classification).*** *Let $\mathbf{a}^k = (\mathbf{x}^k, \mathbf{r}^k, w^k)$ denote the action at step k, where $\mathbf{x}^k$ and $\mathbf{r}^k$ are the translational and rotational action increments, respectively. Given thresholds $\tau_{\text{trans}}, \tau_{\text{rot}}, \tau_w > 0$, we define the hard primitive classifier $\Phi : \mathbf{a}^k \mapsto y^k \in \{0, \dots, 7\}$ as*

$$\Phi(\mathbf{a}^k) = \begin{cases} 0, & \|\mathbf{x}^k\| < \tau_{\text{trans}},\ \|\mathbf{r}^k\| < \tau_{\text{rot}},\ |\Delta w^k| < \tau_w, \quad (\textit{idle}) \\ 1, & -\Delta w^k \geq \tau_w, \quad (\textit{grasp}) \\ 2, & \Delta w^k \geq \tau_w, \quad (\textit{release}) \\ 3, & \|\mathbf{x}^k\| \geq \tau_{\text{trans}},\ \|\mathbf{r}^k\| < \tau_{\text{rot}},\ |\Delta w^k| < \tau_w,\ x_1^k \geq 0, \quad (\textit{push}) \\ 4, & \|\mathbf{x}^k\| \geq \tau_{\text{trans}},\ \|\mathbf{r}^k\| < \tau_{\text{rot}},\ |\Delta w^k| < \tau_w,\ x_1^k < 0, \quad (\textit{pull}) \\ 5, & \|\mathbf{x}^k\| < \tau_{\text{trans}},\ \|\mathbf{r}^k\| \geq \tau_{\text{rot}},\ |\Delta w^k| < \tau_w, \quad (\textit{rotation}) \\ 6, & \|\mathbf{x}^k\| \geq \tau_{\text{trans}},\ \|\mathbf{r}^k\| \geq \tau_{\text{rot}},\ |\Delta w^k| < \tau_w,\ x_1^k \geq 0, \quad (\textit{push+rotation}) \\ 7, & \|\mathbf{x}^k\| \geq \tau_{\text{trans}},\ \|\mathbf{r}^k\| \geq \tau_{\text{rot}},\ |\Delta w^k| < \tau_w,\ x_1^k < 0, \quad (\textit{pull+rotation}) \end{cases} \tag{1}$$

where $\Delta w^k = w^k - w^{k-1}$. The eight cases form an exhaustive and mutually exclusive partition under the above threshold rules.

Definition 1 is formulated in a robot-centric action frame to provide a simple and consistent partition of 7-DoF manipulation actions. The resulting taxonomy is not intended to enumerate all semantic manipulation skills, but rather to capture a compact set of dominant local motion patterns that can serve as structured priors for test-time guidance. Equation (1) enables automatic generation of primitive labels, thus allowing us to learn a primitive classifier without manual annotation. We then train a classifier PANet: $\hat{\mathbf{y}} = \text{PANet}(\mathbf{O})$, which predicts primitive distributions from observations. PANet adopts DINOv2 Oquab et al. (2024) as the visual encoder and T5 Raffel et al. (2020) as the text encoder, followed by a multimodal fusion module and an MLP classification head.

Importantly, PANet predicts probabilistic primitive distributions rather than hard labels. This probabilistic formulation improves robustness under ambiguous transitions between primitives and naturally supports multi-primitive guidance during long-horizon manipulation.

3.2 PRIMITIVE GUIDANCE

To enable differentiable guidance while preserving local consistency with the corresponding hard primitive assignments, we define a soft primitive distribution induced by the generated actions.

Definition 2. ***(Differentiable Soft Primitive Classification).*** *Let $\Delta w^k = w^k - w^{k-1}$, $d_{\text{trans}}^k = \|\mathbf{x}^k\|$, $d_{\text{rot}}^k = \|\mathbf{r}^k\|$. To construct a differentiable relaxation of the hard classifier in Definition 1, we replace each binary condition with a smooth gate. Given a smoothing parameter $\beta > 0$, define*

$$\sigma_\beta(z) = \sigma\left(\frac{z}{\beta}\right) = \frac{1}{1 + \exp(-z/\beta)}.$$

We introduce the following gates:

$$g_{\text{trans}}^+ = \sigma_\beta\left(d_{\text{trans}}^k - \tau_{\text{trans}}\right),\ g_{\text{trans}}^- = \sigma_\beta\left(\tau_{\text{trans}} - d_{\text{trans}}^k\right),\ g_{\text{rot}}^+ = \sigma_\beta\left(d_{\text{rot}}^k - \tau_{\text{rot}}\right),$$
$$g_{\text{rot}}^- = \sigma_\beta\left(\tau_{\text{rot}} - d_{\text{rot}}^k\right),\ g_w^0 = \sigma_\beta\left(\tau_w - |\Delta w^k|\right),\ g_{\text{grasp}} = \sigma_\beta\left(-\Delta w^k - \tau_w\right),$$
$$g_{\text{release}} = \sigma_\beta\left(\Delta w^k - \tau_w\right),\ g_{\text{push}} = \sigma_\beta(x_1^k),\ g_{\text{pull}} = \sigma_\beta(-x_1^k).$$

For each primitive j, we define a soft compatibility score by taking the geometric mean of the gates corresponding exactly to the conditions in Definition 1:

$$q_0 = \left(g_{\text{trans}}^- g_{\text{rot}}^- g_w^0\right)^{1/3},\ q_2 = g_{\text{release}},\ q_4 = \left(g_{\text{trans}}^+ g_{\text{rot}}^- g_w^0 g_{\text{pull}}\right)^{1/4},\ q_6 = \left(g_{\text{trans}}^+ g_{\text{rot}}^+ g_w^0 g_{\text{push}}\right)^{1/4},$$
$$q_1 = g_{\text{grasp}},\ q_3 = \left(g_{\text{trans}}^+ g_{\text{rot}}^- g_w^0 g_{\text{push}}\right)^{1/4},\ q_5 = \left(g_{\text{trans}}^- g_{\text{rot}}^+ g_w^0\right)^{1/3},\ q_7 = \left(g_{\text{trans}}^+ g_{\text{rot}}^+ g_w^0 g_{\text{pull}}\right)^{1/4}.$$

The differentiable primitive distribution induced by the action is then defined as

$$p_j^k = \frac{(q_j)^{1/\tau}}{\sum_{i=0}^{7}(q_i)^{1/\tau}}, \qquad j \in \{0, \dots, 7\}, \tag{2}$$

where $\tau > 0$ controls the sharpness of the primitive distribution. For actions that do not lie exactly on a decision boundary, the soft classifier is consistent with the hard classifier:

$$\lim_{\beta \to 0} p^k_{\Phi(\mathbf{a}^k)} = 1, \qquad \lim_{\beta \to 0} p_j^k = 0, \quad j \neq \Phi(\mathbf{a}^k). \tag{3}$$

Therefore, Definition 2 provides a differentiable relaxation of the hard primitive in Definition 1.

Given predicted primitive distributions $\hat{\mathbf{y}}^k$ and generated actions $\mathbf{a}^k$, the primitive guidance is:

$$\mathcal{L}_{PG}(\mathbf{a}^k, \mathbf{O}^k) = H(\hat{\mathbf{y}}^k, \mathbf{p}^k) = H(\text{PANet}(\mathbf{O}^k), \mathbf{p}^k)\,, \tag{4}$$

where $H(\cdot, \cdot)$ denotes cross-entropy between predicted primitive distributions and action-induced primitive distributions. Unlike hard primitive constraints or single-label guidance, this probabilistic formulation enables smooth transitions between primitives and improves execution robustness. In particular, multiple primitives may simultaneously receive non-negligible probabilities during transitional stages, such as switching from `push` to `rotation`. This soft guidance reduces abrupt action corrections and improves trajectory consistency.

3.3 Diffusion Policy with Primitives Guidance

Diffusion policy formulates the manipulation process as a Markov process and leverages DDPM Ho et al. (2020) to model the manipulation policy with multimodal observations $\mathbf{O}^k$. Given observation $\mathbf{O}^k$, it performs a sequence of T denoising steps starting from a random action $\mathbf{a}_T^k \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to generate an action $\mathbf{a}_0^k$ with iteration formula $\mathbf{a}_{t-1}^k = \alpha_t(\mathbf{a}_t^k - \gamma_t \varepsilon_\theta(\mathbf{O}^k, \mathbf{a}_t^k, t) + \epsilon)$, where $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma_t^2 \mathbf{I})$. The coefficients $\alpha_t, \gamma_t, \sigma_t$ are predefined noise schedules of the denoising step t.

To improve the transferability of diffusion policies, we incorporate the primitive guidance mechanism introduced in Section 3.2 into the denoising process. By leveraging test-time observations to refine generated actions, the proposed approach steers denoising updates toward primitive-consistent trajectories, resulting in more robust and semantically coherent action generation under distribution shifts. This yields the following guided diffusion policy:

$$\begin{aligned}
\tilde{\mathbf{a}}_{t-1}^k &= \alpha_t(\mathbf{a}_t^k - \gamma_t \varepsilon_\theta(\mathbf{O}^k, \mathbf{a}_t^k, t) + \epsilon)\,, \\
\hat{\mathbf{a}}_0^k &= (\tilde{\mathbf{a}}_{t-1}^k - \sqrt{1-\bar{\alpha}_t}\varepsilon_\theta(\mathbf{O}^k, \mathbf{a}_t^k, t))/\sqrt{\bar{\alpha}_t}\,, \\
\mathbf{a}_{t-1}^k &= \tilde{\mathbf{a}}_{t-1}^k - \eta \nabla_{\mathbf{a}_0^k} \mathcal{L}_{\text{PG}}(\hat{\mathbf{a}}_0^k, \mathbf{O}^k)\,,
\end{aligned} \tag{5}$$

Here, $\hat{\mathbf{a}}_0^k$ denotes the estimated noise-free action obtained through the pre-defined inverse diffusion schedule. By incorporating the primitive guidance term $\nabla_{\mathbf{a}} \mathcal{L}_{\text{PG}}(\cdot, \cdot)$, Equation (5) constrains the update direction of the inverse diffusion, ensuring consistency with the intended primitive actions.

3.4 Flow Policy with Primitives Guidance

Flow policies leverage flow matching to learn a continuous transformation from noisy to expert actions. Given an action $\mathbf{a}$, the flow path was defined as $\mathbf{a}_t = \alpha_t \mathbf{a} + b_t \mathbf{z}, t \in [0, 1]$, where α_t and b_t are pre-defined schedules, $\mathbf{z} \sim p(\mathbf{z})$ is an initial noise action. A commonly used schedule is $\alpha_t = t$ and $b_t = 1 - t$, then the velocity field is $\mathbf{v}(\mathbf{a}_t, t) = \frac{\mathrm{d}\mathbf{a}_t}{\mathrm{d}t} = \mathbf{a} - \mathbf{z}$. The flow policy, parameterized by θ, predicts velocity field $\mathbf{v}_\theta(\mathbf{a}_t, \mathbf{O}, t)$. After training, it generates $\mathbf{a}_1$ from noise $\mathbf{a}_0 = \mathbf{z}$ by using forward Euler rule $\mathbf{a}_{t+\delta} = \mathbf{a}_t + \delta \mathbf{v}_\theta(\mathbf{a}_t, \mathbf{O}, t)$, where δ is the integration step size.

Similarly, we integrate the primitive guidance in Section 3.2 into the inference phase of flow policies. Specifically, we leverage the primitives derived from test-time observation to constrain the continuous actions generated, thereby yielding actions that are consistent with the target primitives. From the pre-defined flow schedule, we can deduce the following guided flow policy:

$$\begin{aligned}
\tilde{\mathbf{a}}_{t+\delta}^k &= \mathbf{a}_t^k + \delta \mathbf{v}_\theta(\mathbf{a}_t^k, \mathbf{O}^k, t), \\
\hat{\mathbf{a}}_1^k &= \mathbf{a}_t^k + (1-t)\mathbf{v}_\theta(\mathbf{a}_t^k, \mathbf{O}^k, t), \\
\mathbf{a}_{t+\delta}^k &= \tilde{\mathbf{a}}_{t+\delta}^k - \eta \nabla_{\mathbf{a}_1^k} \mathcal{L}_{\text{PG}}(\hat{\mathbf{a}}_1^k, \mathbf{O}^k).
\end{aligned} \tag{6}$$

This is equivalent to defining a guided velocity field $\mathbf{v}_\theta^{\text{guided}} = \mathbf{v}_\theta - \frac{\eta}{\delta}\nabla_{\mathbf{a}_1^k}\mathcal{L}_{\text{PG}}(\hat{\mathbf{a}}_1, \mathbf{O})$.

Our method incorporates primitive constraints into the action denoising process of diffusion- and flow-based policies, steering generated actions toward the desired manipulation primitives. Moreover, under local alignment and smoothness conditions, we show that a primitive-guidance update admits a contraction-style bound on the deviation from a reference action, up to a residual term that captures primitive-guidance mismatch. The formal result and its proof are provided in Appendix A.

4 Experiments

We design experiments to answer the following questions: **i)** How accurate and robust is PANet for primitive prediction? **ii)** Can the proposed primitive guidance improve the manipulation performance of pretrained policies without retraining? **iii)** Does PriGo generalize effectively to real-world robotic manipulation tasks? **iv)** How does each core component contribute to the overall performance? We first evaluate the accuracy and robustness of PANet in Section 4.1. Next, we demonstrate the performance improvements of PriGo on simulated benchmarks in Section 4.2 and real-world robotic tasks in Section 4.3. Finally, we conduct ablation studies in Section 4.4.

4.1 Analysis of PANet Robustness

Type	Idle	Grasp	Rel.	Push	Pull	Rot.	Push+Rot.	Pull+Rot.
Idle	**90**	6	2	1	1	0	0	0
Grasp	2	**95**	0	1	2	0	0	0
Rel.	1	2	**96**	1	0	0	0	0
Push	0	0	0	**94**	2	1	3	0
Pull	0	0	0	1	**95**	1	0	3
Rot.	0	0	0	1	2	**93**	2	2
Push+Rot.	0	0	0	5	0	4	**91**	0
Pull+Rot.	0	0	0	0	6	4	0	**90**

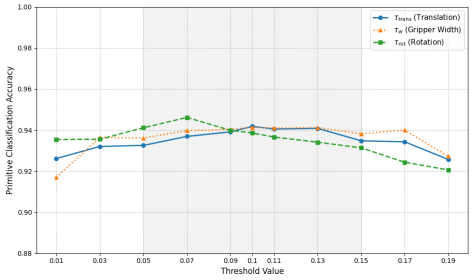


Figure 3: Confusion matrix of PANet primitive classification (left) and sensitivity analysis of auto-labeling thresholds (right). Rel. and Rot. denote release and rotation, respectively.

To evaluate the reliability of PANet, we report its primitive classification accuracy and confusion matrix in the left subfigure of Figure 3. For a clear and balanced visualization, the confusion matrix is computed using 1,000 randomly sampled test examples from each primitive category. On the full test set of LIBERO Liu et al. (2023), PANet achieves an average classification accuracy of 94%. Most errors occur between closely related primitives, such as `push+rotation` and `push` or `rotation`, indicating that the remaining misclassifications are largely concentrated among primitives with similar local motion structures. We further evaluate PANet on SIMPLER Li et al. (2025a), which is also included in the PANet training corpus, and obtain an average accuracy of 95%. For both benchmarks, evaluation is conducted on held-out trajectories that are excluded from PANet training. These results demonstrate that PANet maintains consistently high primitive classification accuracy across both benchmarks.

We further analyze the sensitivity of PriGo to the thresholds τ_{trans}, τ_{rot}, and τ_w. As shown in the right subfigure of Figure 3, performance remains stable within a broad range of threshold values, demonstrating that PriGo is not overly sensitive to specific hyperparameter value choices.

Table 1: Success rate on LIBERO with four settings (spatial, object, goal, long-horizon). Comparison between with (w/ PG) and without (w/o PG) primitive guidance. Here π_0+PG and SmolVLA+PG are PriGo-FP. DP+PG and CogACT+PG are PriGo-DP.

Methods	Spatial		Object		Goal		Long		Avg.	
	w/o PG	w/ PG	w/o PG	w/ PG	w/o PG	w/ PG	w/o PG	w/ PG	w/o PG	w/ PG
π_0 Black et al. (2024a)	89.2	91.0	84.6	87.8	94.4	95.2	75.2	78.8	85.6	88.2
SmolVLA	92.0	93.6	93.4	94.4	91.4	93.2	75.8	79.8	88.2	90.3
DP Chi et al. (2023)	76.2	80.6	87.4	90.8	70.4	72.6	54.2	58.4	72.1	75.6
CogACT Li et al. (2024)	91.1	95.6	92.0	97.3	91.5	95.8	76.4	80.4	87.8	92.3

4.2 SIMULATED EVALUATION

Evaluation on LIBERO. LIBERO Liu et al. (2023) is a large-scale benchmark encompassing over 130 tasks and categorized into five distinct suites. To verify the effectiveness of the proposed primitive guidance, we integrate it into a suite of pretrained policies, including the flow policies π_0 Black et al. (2024a), SmolVLA Shukor et al. (2025), diffusion policies DP Chi et al. (2023), and CogACT Li et al. (2024). This integration yields two policy variants, namely PriGo-DP and PriGo-FP. For π_0 and SmolVLA, we adopt the pre-trained models and hyperparameter configurations provided by LeRobot[1]. As for DP and CogAct, no official pre-trained checkpoints on LIBERO are available; we thus fine-tuned these two models on LIBERO.

Table 1 presents the evaluation results of four pretrained policies, with and without the integration of our primitive guidance module. As shown, PriGo consistently improves performance across both flow- and diffusion-based policies, regardless of whether the backbone is a large-scale model (e.g., π_0 and CogACT) or a lightweight DP. The improvements are also consistently observed across all four LIBERO suites, including the challenging long-horizon setting, suggesting that the benefits of primitive guidance are not specific to a particular policy architecture or task category. By encouraging generated actions to remain consistent with the predicted local motion structure during inference, PriGo provides a general mechanism for refining actions from heterogeneous generative policies without modifying or retraining their backbone models. These results suggest that primitive-consistent action refinement can improve execution robustness across different policy architectures while preserving the plug-and-play nature of the underlying framework.

Evaluation on SIMPLER. SIMPLER benchmark Li et al. (2025a) provides two evaluation settings: *Visual Matching*, which closely mirrors real-world scenarios by minimizing the gap between simulated and physical environments, and *Variant Aggregations*, which introduces variations in background, lighting, distractors, and table textures. We compare our PriGo-DP with five other VLA models, with the corresponding results summarized in Table 2. Notably, all baseline results presented in this table are from CogACT. As shown, PriGo-DP outperforms all existing baseline methods by a clear margin. Specifically, relative to CogACT, our approach delivers an improvement of 5.1 absolute percentage points under the Visual Matching setting and 7.5 percentage points under the Variant Aggregation setting. These results demonstrate that PriGo can effectively enhance the test-time performance of the VLA model CogACT in a plug-and-play manner, without the need for additional retraining.

Table 2: Comparison of our approach PriGo-DP (CogACT with PriGo) with existing VLA models on the Google robot across four tasks in two SIMPLER settings.

Google Robot	Method	Pick Coke Can	Move Near	Open/Close Drawer	Open Top Drawer and Place Apple	*Average*
SIMPLER (Visual Matching)	RT-2-X Vuong et al. (2023)	78.7	77.9	25.0	3.7	46.3
	Octo-Base Team et al. (2024)	17.0	4.2	22.7	0.0	11.0
	OpenVLA Kim et al. (2025)	18.0	56.3	63.0	0.0	34.3
	CogACT Li et al. (2024)	91.3	85.0	71.8	50.9	74.8
	PriGo-DP	**93.2**	**89.3**	**77.5**	**59.7**	**79.9**
SIMPLER (Variant Aggregation)	RT-2-X Vuong et al. (2023)	82.3	79.2	35.3	20.6	54.4
	Octo-Base Team et al. (2024)	0.6	3.1	1.1	0.0	1.2
	OpenVLA Kim et al. (2025)	60.8	67.7	28.8	0.0	39.3
	CogACT Li et al. (2024)	89.6	80.8	28.3	46.6	61.3
	PriGo-DP	**92.5**	**83.6**	**44.1**	**55.0**	**68.8**

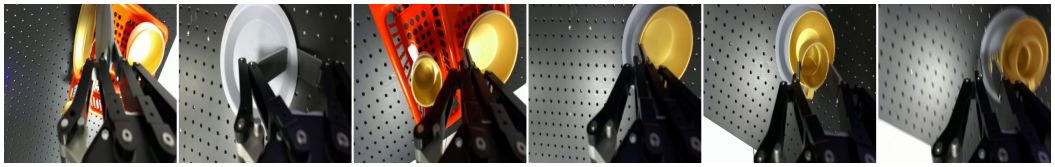
Evaluation on CALVIN. CALVIN Mees et al. (2022) is a benchmark centered on a Franka Panda robot interacting within richly parameterized tabletop environments. It has four environments that differ in desk textures and object placements. The benchmark defines 34 tasks with sequential execution of language-conditioned instructions, where each instruction comprises five steps. We integrate the proposed primitive guidance into the pretrained diffusion policy 3D Diffuser Actor (3DDA) Ke et al. (2024), and compare its performance with that of several SoTA methods on the CALVIN, including DTP Hou et al. (2024), DP3 Ze et al. (2024), 3DDA, and ADPro Li et al. (2025c). The results are reported in Table 3.

[1] https://github.com/huggingface/lerobot

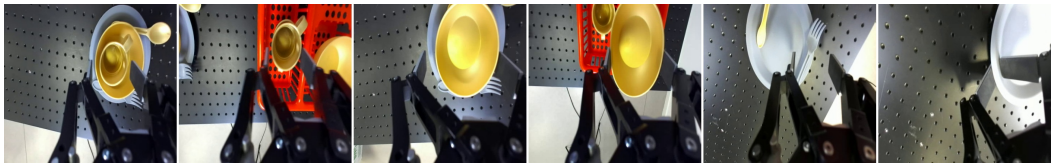
As shown in Table 3, our PriGo-DP (3DDA + PriGo), outperforms other diffusion policies by a notable margin. Specifically, PriGo-DP increases the average length compared to the vanilla diffusion policy 3DDA by 0.83, enhancing its performance on zero-shot long-horizon tasks. These results demonstrate that PriGo can effectively boost test-time generalization, even for non-VLA models.

Table 3: Zero-shot (Train A, B, C →Test D) long-horizon evaluation on CALVIN. Avg. Len denotes the average length of correctly predicted trajectories.

Method	Task completed in a row 1	2	3	4	5	Avg. Len ↑
DTP	94.5	82.5	72.8	61.3	50.0	3.61
DP3	53.9	44.7	38.0	34.3	29.0	2.00
3DDA	93.8	80.3	66.2	53.3	41.2	3.35
ADPro	94.7	83.0	73.6	61.4	51.1	3.64
PriGo-DP	98.1	92.1	87.6	80.4	59.9	4.18



(a) Rollout for the *"Prepare a set of tableware* task".



(b) Rollout for the *"Bring back a set of tableware to the basket"* task.



(c) Rollout for the *"Prepare two sets of tableware"* task.

Figure 4: Rollouts of our PriGo-DP on three table-service tasks. PriGo-DP aligns the generated actions with task primitives, thereby improving policy performance on real-world tasks.

4.3 Real-Robot Evaluation

In our real-world experiments, we use a Franka Emika Panda robot equipped with two RGB cameras. We design three sequential table-service tasks in restaurant-style scenarios: 1) *prepare one set of tableware*, 2) *return one set of tableware to the basket*, and 3) *prepare two sets of tableware*. The tableware consists of a plate, a bowl, a cup, a fork, and a spoon. Each task is evaluated over 10 episodes, and we report the mean ± standard deviation of the episode-level subtask success rates in Table 4. The first two tasks consist of 10 subtasks per episode (five objects, each requiring a pick and a place), while the third consists of 20 subtasks per episode. Accordingly, each episode-level success rate is computed as the percentage of successfully completed subtasks within that episode, and the reported mean is averaged over the 10 evaluation episodes. Equivalently, since all episodes within each task contain the same number of subtasks, each successful subtask contributes 1 percentage point to the mean success rate for the first two tasks (100 subtasks in total) and 0.5 percentage points for the third task (200 subtasks in total). With at most 20 training demonstrations per task, PriGo-DP achieves a success rate of up to 81%. Compared with vanilla CogACT, PriGo-DP yields an average improvement of 26 absolute percentage points across the three tasks. These results demonstrate substantial improvements on long-horizon real-world manipulation tasks.

To further illustrate the real-world performance of PriGo-DP, we present representative rollouts for all three tasks in Figure 4. Each subplot highlights the key sequential steps of a task, from the initial observation to the final goal state. These qualitative results illustrate how PriGo-DP generates coherent sequences of manipulation actions to accomplish long-horizon real-world tasks.

4.4 Ablation Studies

The results in Tables 1 to 4 demonstrate the applicability of PriGo as a plug-and-play guidance mechanism across different policy architectures and evaluation settings.

Table 4: Success rate of PriGo-DP (CogACT with PriGo) on three long-horizon real-world tasks. w/o PG and w/ PG mean with and without PriGo.

Task	#Train	w/o PG	w/ PG
prepare one tableware	12	47 ± 9.3	76 ± 7.3
bring back tableware	18	55 ± 11.3	81 ± 9.6
prepare two tableware	20	40 ± 8.3	62 ± 8.0

Table 5: Ablation study. The policy learned with full 8 primitives outperforms its counterparts that do not distinguish "push", "pull" (PP), and do not use hybrid primitives (HP).

Dataset	w/o PP	w/o HP	FULL
LIBERO	89.9	91.0	92.3
SIMPLER	77.1	77.8	79.9

The proposed primitive guidance can operate in a plug-and-play manner—an attribute that has been validated by the experimental results presented in Tables 1 to 4. To further dissect the efficacy of individual components of primitives, we conduct ablation studies on three distinct schemes: 1) Non-discriminated translation primitives (i.e., treating "push" and "pull" as a unified translational primitive type); 2) No hybrid primitives (excluding "push+rotation" and "pull+rotation"); 3) A full set of eight primitive actions. These results in Table 5 demonstrate that distinguishing directional and hybrid primitives contributes to the effectiveness of primitive-guided action refinement.

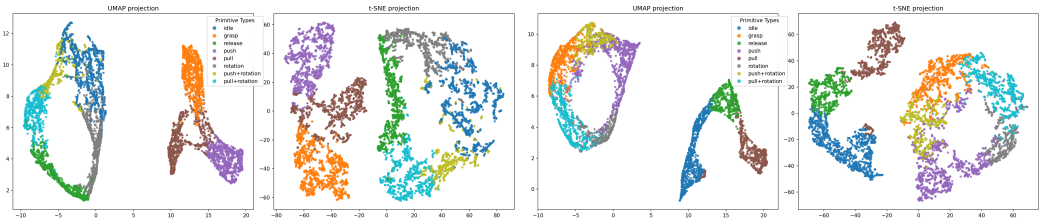


Figure 5: Visualization of actions and primitive labels from two episodes via UMAP and t-SNE.

As shown in Table 5, omitting the "push" and "pull" (PP) types results in the most significant performance degradation of the policy. Without differentiating between push and pull, the generated actions tend to deviate in direction, directly leading to task failure. The model also experiences a performance drop when hybrid primitives are removed. Notably, our observations show that integrating hybrid primitives leads to smoother action outputs from the policy.

Furthermore, to intuitively illustrate the primitive action classification results, we visualize the primitive action instances in Figure 5 using UMAP and t-SNE. The first two subfigures are from the Object subset of LIBERO with 7281 actions, and the third and fourth are from the Spatial subset with 5796 actions. Here, distinct colors denote different categories generated via Equation (1). As evidenced by the results, the same primitive exhibits clear clustering after dimensionality reduction.

5 Conclusion

In this paper, we introduce PriGo, a primitive guidance approach that can work with both lightweight policies and large VLA models. PriGo effectively decouples discrete primitive planning from continuous action learning. To capture primitive planning for a given task and its visual observations, we propose a primitive classification module and an automatic primitive labeling method, enabling primitive prediction directly from current observations. To ensure that the generated actions align with the predicted primitives, we introduced a differentiable primitive guidance mechanism, which constrains action generation, leading to more robust manipulation. This guidance can be seamlessly integrated into pretrained diffusion and flow policies, offering a flexible, plug-and-play solution that enhances performance without the need for retraining. Experimental results demonstrate that our policy outperforms baseline methods in both simulation and real-world tasks.

Limitations. Although PriGo improves both flow-based and diffusion-based policies at test time, the gains remain limited. This mainly results from two factors. First, the plug-and-play design enables deployment without retraining but inherently restricts performance improvements. Future work could explore tighter integration with end-to-end training. Second, PANet's generalization is still constrained by model size and training data scale. More suitable data and incorporating historical robot states may further improve the effectiveness of the guidance mechanism.

AI USE STATEMENT

In this work, we used generative AI tools for paper polishing, including improving grammar, clarity, readability, and presentation of author-written text. We did not use generative AI tools for generating research ideas, formulating hypotheses, conducting experiments, analyzing or interpreting results, or producing scientific claims. Other disclosure tasks not relevant to this work are not applicable.

All AI-assisted edits were reviewed by the authors to ensure that they accurately preserved the intended technical meaning and did not introduce unsupported claims, incorrect information, or inappropriate citations. The authors independently verified the technical content and retained full control over the final manuscript.

We take responsibility for the final content of this work, including any text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects or the collection of personally identifiable or sensitive information. We are not aware of any significant ethical concerns arising from the methodology or experiments presented in this work. We have conducted this research in accordance with standard principles of research integrity and have reported our methodology and results as accurately and transparently as possible.

REPRODUCIBILITY STATEMENT

We provide the methodological, implementation, and experimental details necessary to reproduce the main results of this work. The primitive taxonomy, automatic labeling procedure, differentiable primitive guidance objective, and its integration with diffusion- and flow-based policies are specified in Sec. 3. Appendix B provides implementation details of PANet, including its architecture, training objective, optimizer, hyperparameters, and computational resources. The benchmark settings, backbone policies, and evaluation protocols are described in Sec. 4, while additional ablations, hyperparameter sensitivity analyses, qualitative results, and inference-efficiency measurements are provided in Appendix C. The assumptions and derivations used in our theoretical analysis are documented in Appendix A. We additionally provide an anonymized implementation containing the primitive labeling and guidance modules, PANet training code, evaluation scripts, and experiment configurations to facilitate independent reproduction of our results.

REFERENCES

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

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, 2024. *arXiv preprint arXiv:2410.24164*, 2024a.

Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Rich Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pre-trained image-editing diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024b.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y Galliker, et al. $pi_{0.5}$: a vision-language-action model with open-world generalization. In *9th Annual Conference on Robot Learning*, 2025.

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

Murtaza Dalal, Deepak Pathak, and Russ R Salakhutdinov. Accelerating robotic reinforcement learning via parameterized action primitives. *Advances in Neural Information Processing Systems*, 34:21847–21859, 2021.

Shengliang Deng, Mi Yan, Songlin Wei, Haixin Ma, Yuxin Yang, Jiayi Chen, Zhiqi Zhang, Taoyu Yang, Xuheng Zhang, Heming Cui, et al. Graspvla: a grasping foundation model pre-trained on billion-scale synthetic action data. *arXiv preprint arXiv:2505.03233*, 2025.

Niklas Funk, Julen Urain, Joao Carvalho, Vignesh Prasad, Georgia Chalvatzaki, and Jan Peters. Actionflow: Efficient, accurate, and fast policies with spatially symmetric flow matching. In *R: SS workshop: Structural Priors as Inductive Biases for Learning Robot Dynamics*, 2024.

Tian Gao, Soroush Nasiriany, Huihan Liu, Quantao Yang, and Yuke Zhu. Prime: Scaffolding manipulation tasks with behavior primitives for data-efficient imitation learning. *IEEE Robotics and Automation Letters*, 2024.

Ricardo Garcia, Shizhe Chen, and Cordelia Schmid. Towards generalizable vision-language robotic manipulation: A benchmark and llm-guided 3d policy. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8996–9002. IEEE, 2025.

Zhihao Gu, Ming Yang, Difan Zou, and Dong Xu. Learning diffusion policy from primitive skills for robot manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18262–18270, 2026.

Yijie Guo, Bingjie Tang, Iretiayo Akinola, Dieter Fox, Abhishek Gupta, and Yashraj Narang. Srsa: Skill retrieval and adaptation for robotic assembly tasks. In *The Thirteenth International Conference on Learning Representations*, 2025.

Haoran He, Chenjia Bai, Ling Pan, Weinan Zhang, Bin Zhao, and Xuelong Li. Learning an actionable discrete diffusion policy via large-scale actionless video pre-training. *Advances in Neural Information Processing Systems*, 37:31124–31153, 2025.

Ayano Hiranaka, Minjune Hwang, Sharon Lee, Chen Wang, Li Fei-Fei, Jiajun Wu, and Ruohan Zhang. Primitive skill-based robot learning from human evaluative feedback. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7817–7824. IEEE, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Zhi Hou, Tianyi Zhang, Yuwen Xiong, Hengjun Pu, Chengyang Zhao, Ronglei Tong, Yu Qiao, Jifeng Dai, and Yuntao Chen. Diffusion transformer policy. *arXiv preprint arXiv:2410.15959*, 2024.

Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. In *8th Annual Conference on Robot Learning*, 2024.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, et al. Openvla: An open-source vision-language-action model. In *Conference on Robot Learning*, pp. 2679–2713. PMLR, 2025.

Woo Kyung Kim, Youngseok Lee, Jooyoung Kim, and Honguk Woo. Llm-based skill diffusion for zero-shot policy adaptation. *Advances in Neural Information Processing Systems*, 37:6749–6775, 2024.

Moritz Knaust and Dorothea Koert. Guided robot skill learning: A user-study on learning probabilistic movement primitives with non-experts. In *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, pp. 514–521. IEEE, 2021.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv:2411.19650*, 2024.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Oier Mees, Karl Pertsch, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. In *Conference on Robot Learning*, pp. 3705–3728. PMLR, 2025a.

Zezeng Li, Alexandre Chapin, Enda Xiang, Rui Yang, Bruno Machado, Na Lei, Emmanuel Dellandrea, Di Huang, and Liming Chen. Robotic manipulation via imitation learning: Taxonomy, evolution, benchmark, and challenges. *arXiv preprint arXiv:2508.17449*, 2025b.

Zezeng Li, Rui Yang, Ruochen Chen, ZhongXuan Luo, and Liming Chen. Adpro: a test-time adaptive diffusion policy for robot manipulation via manifold and initial noise constraints. *arXiv preprint arXiv:2508.06266*, 2025c.

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

Jiaming Liu, Hao Chen, Pengju An, Zhuoyang Liu, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, Mengzhen Liu, et al. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv:2503.10631*, 2025a.

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

Tenglong Liu, Jianxiong Li, Yinan Zheng, Haoyi Niu, Yixing Lan, Xin Xu, and Xianyuan Zhan. Skill expansion and composition in parameter space. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Kangchen Lv, Mingrui Yu, Yongyi Jia, Chenyu Zhang, and Xiang Li. Kinematics-aware diffusion policy with consistent 3d observation and action space for whole-arm robotic manipulation. *IEEE Robotics and Automation Letters*, 2026.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022.

J Merel, L Hasenclever, A Galashov, A Ahuja, V Pham, G Wayne, Y Teh, and N Heess. Neural probabilistic motor primitives for humanoid control. In *International Conference on Learning Representations*, 2019.

Fei Ni, Jianye Hao, Yao Mu, Yifu Yuan, Yan Zheng, Bin Wang, and Zhixuan Liang. Metadiffuser: Diffusion model as conditional planner for offline meta-rl. In *International Conference on Machine Learning*, pp. 26087–26105. PMLR, 2023.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research Journal*, pp. 1–31, 2024.

Mingjie Pan, Jiyao Zhang, Tianshu Wu, Yinghao Zhao, Wenlong Gao, and Hao Dong. Omnimanip: Towards general robotic manipulation via object-centric interaction primitives as spatial constraints. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 17359–17369, 2025.

Xue Bin Peng, Michael Chang, Grace Zhang, Pieter Abbeel, and Sergey Levine. Mcp: Learning composable hierarchical control with multiplicative compositional policies. *Advances in neural information processing systems*, 32, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL `http://jmlr.org/papers/v21/20-074.html`.

Quentin Rouxel, Andrea Ferrari, Serena Ivaldi, and Jean-Baptiste Mouret. Flow matching imitation learning for multi-support manipulation. In *2024 IEEE-RAS 23rd International Conference on Humanoid Robots (Humanoids)*, pp. 528–535. IEEE, 2024.

Ranjan Sapkota, Yang Cao, Konstantinos I Roumeliotis, and Manoj Karkee. Vision-language-action models: Concepts, progress, applications and challenges. *arXiv preprint arXiv:2505.04769*, 2025.

Lucy Xiaoyang Shi, Brian Ichter, Michael Equi, Liyiming Ke, Karl Pertsch, Quan Vuong, James Tanner, Anna Walling, Haohuan Wang, Niccolo Fusai, et al. Hi robot: Open-ended instruction following with hierarchical vision-language-action models. *arXiv preprint arXiv:2502.19417*, 2025.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv:2506.01844*, 2025.

Qiao Sun, Liujia Yang, Wei Tang, Wei Huang, Kaixin Xu, Yongchao Chen, Mingyu Liu, Jiange Yang, Haoyi Zhu, Yating Wang, et al. Learning primitive embodied world models: Towards scalable robotic learning. *arXiv preprint arXiv:2508.20840*, 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:2405.12213*, 2024.

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

Weikang Wan, Yifeng Zhu, Rutav Shah, and Yuke Zhu. Lotus: Continual imitation learning for robot manipulation through unsupervised skill discovery. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 537–544, 2024.

Dian Wang, Stephen Hart, David Surovik, Tarik Kelestemur, Haojie Huang, Haibo Zhao, Mark Yeatman, Jiuguang Wang, Robin Walters, and Robert Platt. Equivariant diffusion policy. *8th Annual Conference on Robot Learning*, 2024.

Yuanfei Wang, Xiaojie Zhang, Ruihai Wu, Yu Li, Yan Shen, Mingdong Wu, Zhaofeng He, Yizhou Wang, and Hao Dong. Adamanip: Adaptive articulated object manipulation environments and policy learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Junjie Wen, Minjie Zhu, Yichen Zhu, Zhibin Tang, Jinming Li, Zhongyi Zhou, Chengmeng Li, Xiaoyu Liu, Yaxin Peng, Chaomin Shen, et al. Diffusion-vla: Scaling robot foundation models via unified diffusion and autoregression. *arXiv:2412.03293*, 2024.

Zhou Xian, Nikolaos Gkanatsios, Theophile Gervet, Tsung-Wei Ke, and Katerina Fragkiadaki. Chaineddiffuser: Unifying trajectory diffusion and keypose prediction for robotic manipulation. In *Conference on Robot Learning*, pp. 2323–2339. PMLR, 2023.

Enda Xiang, Haoxiang Ma, Xinzhu Ma, Zicheng Liu, and Di Huang. Graspldp: Towards generalizable grasping policy via latent diffusion. *arXiv preprint arXiv:2602.22862*, 2026.

Jingyun Yang, Ziang Cao, Congyue Deng, Rika Antonova, Shuran Song, and Jeannette Bohg. Equibot: Sim (3)-equivariant diffusion policy for generalizable and data efficient learning. In *8th Annual Conference on Robot Learning*, 2024.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *ICRA 2024 Workshop on 3D Visual Representations for Robot Manipulation*, 2024.

Fan Zhang and Michael Gienger. Affordance-based robot manipulation with flow matching. *arXiv preprint arXiv:2409.01083*, 2024.

Jesse Zhang, Jiahui Zhang, Karl Pertsch, Ziyi Liu, Xiang Ren, Minsuk Chang, Shao-Hua Sun, and Joseph J Lim. Bootstrap your own skills: Learning to solve new tasks with large language model guidance. In *7th Annual Conference on Robot Learning*, 2023.

Zhongyi Zhou, Yichen Zhu, Minjie Zhu, Junjie Wen, Ning Liu, Zhiyuan Xu, Weibin Meng, Yaxin Peng, Chaomin Shen, Feifei Feng, et al. Chatvla: Unified multimodal understanding and robot control with vision-language-action model. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5377–5395, 2025.

This document presents the error-bound proposition, along with implementation details for PANet and a primitive-action analysis. It also presents additional experimental results, including comparisons of sensitivity analysis of the guidance strength η, primitive-level ablations, qualitative comparisons on real-world robotic tasks, and an analysis of inference efficiency.

A Error Bound

Proposition 1. ***(Local Error Bound under Primitive Guidance).*** *Let $\tilde{\mathbf{a}}$ denote an action proposed by the pretrained backbone policy and let $\mathbf{a}^\star$ denote a reference action. Consider one primitive-guidance update*

$$\mathbf{a}^+ = \tilde{\mathbf{a}} - \eta \mathbf{G}(\tilde{\mathbf{a}}), \qquad \mathbf{G}(\mathbf{a}) := \nabla_{\mathbf{a}} \mathcal{L}_{\mathrm{PG}}(\mathbf{a}),$$

where $\eta > 0$ is the guidance step size. Assume that, in a neighborhood $\mathcal{N}$ of $\mathbf{a}^\star$, the primitive-guidance field satisfies

$$\langle \mathbf{G}(\mathbf{a}) - \mathbf{G}(\mathbf{a}^\star), \mathbf{a} - \mathbf{a}^\star \rangle \geq m \|\mathbf{a} - \mathbf{a}^\star\|_2^2, \tag{7}$$

$$\|\mathbf{G}(\mathbf{a}) - \mathbf{G}(\mathbf{a}^\star)\|_2 \leq L \|\mathbf{a} - \mathbf{a}^\star\|_2, \tag{8}$$

for some $m > 0$ and $L > 0$. Let $\epsilon_{\mathrm{PG}} := \|\mathbf{G}(\mathbf{a}^\star)\|_2$ denote the residual primitive-guidance error at the reference action. Then, for $\tilde{\mathbf{a}} \in \mathcal{N}$ and $0 < \eta < 2m/L^2$, the guided action satisfies

$$\|\mathbf{a}^+ - \mathbf{a}^\star\|_2 \leq \rho \|\tilde{\mathbf{a}} - \mathbf{a}^\star\|_2 + \eta \epsilon_{\mathrm{PG}},$$

where $\rho := \sqrt{1 - 2\eta m + \eta^2 L^2} < 1$. In particular, when the primitive guidance is locally consistent at the reference action, i.e., $\mathbf{G}(\mathbf{a}^\star) = \mathbf{0}$, the update is locally contractive:

$$\|\mathbf{a}^+ - \mathbf{a}^\star\|_2 \leq \rho \|\tilde{\mathbf{a}} - \mathbf{a}^\star\|_2. \tag{9}$$

Proof. Let $\mathbf{e} = \tilde{\mathbf{a}} - \mathbf{a}^\star$ and define $\Delta \mathbf{G} = \mathbf{G}(\tilde{\mathbf{a}}) - \mathbf{G}(\mathbf{a}^\star)$. The guidance update gives

$$\mathbf{a}^+ - \mathbf{a}^\star = \tilde{\mathbf{a}} - \eta \mathbf{G}(\tilde{\mathbf{a}}) - \mathbf{a}^\star = \mathbf{e} - \eta \Delta \mathbf{G} - \eta \mathbf{G}(\mathbf{a}^\star).$$

By the triangle inequality,

$$\|\mathbf{a}^+ - \mathbf{a}^\star\|_2 \leq \|\mathbf{e} - \eta \Delta \mathbf{G}\|_2 + \eta \|\mathbf{G}(\mathbf{a}^\star)\|_2.$$

For the first term,

$$\|\mathbf{e} - \eta \Delta \mathbf{G}\|_2^2 = \|\mathbf{e}\|_2^2 - 2\eta \langle \Delta \mathbf{G}, \mathbf{e} \rangle + \eta^2 \|\Delta \mathbf{G}\|_2^2 \leq \left(1 - 2\eta m + \eta^2 L^2\right) \|\mathbf{e}\|_2^2,$$

where the inequality follows from Eqs. equation 7 and equation 8. Therefore,

$$\|\mathbf{e} - \eta \Delta \mathbf{G}\|_2 \leq \rho \|\mathbf{e}\|_2,$$

where

$$\rho = \sqrt{1 - 2\eta m + \eta^2 L^2}.$$

Since $0 < \eta < 2m/L^2$, we have $\rho < 1$. Finally, using $\epsilon_{\mathrm{PG}} = \|\mathbf{G}(\mathbf{a}^\star)\|_2$, we obtain

$$\|\mathbf{a}^+ - \mathbf{a}^\star\|_2 \leq \rho \|\tilde{\mathbf{a}} - \mathbf{a}^\star\|_2 + \eta \epsilon_{\mathrm{PG}},$$

which proves the result. □

B Details Implementation and Primitives Analysis

This section elaborates on additional details regarding the implementation and training of PANet, the primitive classifier proposed in our work.

PANet takes the observation $\mathbf{O}$ as inputs and outputs the primitive type of the next action, i.e., $y = \text{PANet}(\mathbf{O})$. For implementation, we adopt T5 Raffel et al. (2020) as the text encoder and DINOv2 Oquab et al. (2024) as the image encoder. Subsequently, we utilize the multi-modal causal Q-Former Li et al. (2023) to fuse features from these two modalities, which are then fed into the classification head for class probability output. The Q-Former is configured with 3 layers, an embedding dimension of 1280, a cross-attention dimension of 768, 8 attention heads, and 32 queries. The classification head employs a multi-layer perceptron (MLP) architecture consisting of three MLP layers

interleaved with ReLU activation functions (i.e., MLP → ReLU → MLP → ReLU → MLP). The input and output feature dimensions of the classification head are 1280 and 8, respectively: 1280 corresponds to the concatenated dimension of text and image features, while 8 denotes the number of primitive action types. The middle MLP layer is configured with 256 input and output channels. For detailed implementation details, refer to our publicly available codebase at **PriGo**.

For training, PANet is optimized using cross-entropy loss, where the ground-truth labels are obtained via Eq. (1) in our manuscript. We employ the AdamW optimizer with hyperparameters $\beta_1 = 0.9$, $\beta_2 = 0.99$, and a learning rate of $lr = 5 \times 10^{-5}$. The model was trained on a single NVIDIA A100 GPU in about 1 day, utilizing the full LIBERO dataset, 20 demonstrations per task from SIMPLER, and a 50% random subset of CALVIN episodes in environments A, B, and C.

Statistical analysis of primitive actions. We conducted preliminary experiments to analyze the distribution of primitive types under different settings, and selected a set of eight primitive actions for all experiments based on these statistics. Specifically, we observed that certain composite primitives are extremely rare, e.g., *grasp + push* accounting for only 0.07% across all 10 categories. This leads to a long-tail classification problem, degrading both overall accuracy and performance on infrequent classes. Moreover, over-specifying primitives can introduce ambiguity. For instance, splitting rotation into left and right directions is often unnecessary, as both may be valid in many scenarios (e.g., grasping a handle or a cup). Enforcing a fixed label in such cases reduces policy flexibility and introduces label noise. In addition, finer-grained primitives result in greater variation in primitive distributions across tasks, which weakens cross-task generalization.

For example, when rotation is split into clockwise and counterclockwise directions, the primitive distribution for the task "*pick up the salad dressing and place it in the basket*" becomes: {'0': 14.43%, '1': 5.05%, '2': 4.48%, '3': 8.73%, '4': 12.85%, '5': 10.79%, '6': 2.80%, '7': 14.51%, '8': 6.95%, '9': 12.07%, '10': 7.34%}. For "*pick up the cream cheese and place it in the basket*", it becomes: {'0': 13.27%, '1': 5.14%, '2': 7.72%, '3': 10.86%, '4': 10.86%, '5': 5.79%, '6': 11.60%, '7': 4.23%, '8': 9.10%, '9': 6.64%, '10': 14.79%}. Notably, the distribution of the last categories changes significantly across tasks. In contrast, using the original 8 primitives yields more stable distributions. For the first task: '0': 16.04%, '1': 6.34%, '2': 5.16%, '3': 8.47%, '4': 13.58%, '5': 13.94%, '6': 18.98%, '7': 17.49%, and for the second task: {'0': 13.34%, '1': 7.18%, '2': 7.76%, '3': 10.91%, '4': 10.91%, '5': 16.87%, '6': 14.30%, '7': 18.73%}.

C Additional Results

In this section, we provide additional analyses of PriGo, including the choice of sensitivity to the guidance strength, primitive-level ablations, qualitative visualizations, and inference efficiency on real-robot tasks.

Sensitivity to guidance strength. We next study the sensitivity of PriGo to the guidance strength η in Eq. 5 and Eq. 6. As shown in Table 6, PriGo consistently improves over the unguided setting ($\eta = 0$) across a broad range of η, indicating that the method is not sensitive to a particular choice of guidance strength. Performance remains consistently high for intermediate values and degrades only when the guidance becomes excessively strong. We use $\eta = 0.002$ for all other experiments.

Table 6: Ablation of the guidance strength η in Eq. 5 and Eq. 6.

η	0	0.0005	0.001	0.002	0.004	0.007	0.01	0.0125	0.015
Avg.	87.8	89.4	91.8	92.3	91.9	92.2	91.6	90.6	88.1

Contribution of individual primitives. We further investigate whether the effectiveness of PriGo is dominated by a particular primitive or task stage. To this end, we enable guidance for one primitive type at a time, including grasp, release, push, pull, rotation, and their compositional variants, while leaving the remaining primitives unguided. Table 7 reports the results. Guidance for most individual primitives improves performance over the unguided baseline, with particularly clear gains from grasp, release, and rotation. Moreover, jointly applying guidance to all primitives achieves the highest success rate of 92.3%. These results indicate that the benefit of PriGo is distributed across

different manipulation stages rather than being attributable to a single primitive, and that primitive guidance is most effective when applied throughout the task.

Table 7: Ablation of guidance applied to different primitive actions.

w/	Grasp	Rel.	Push	Pull	Rot.	Push+Rot.	Pull+Rot.	All
Avg.	91.0	90.5	89.4	87.9	90.3	90.6	89.4	92.3

Qualitative analysis. To better understand how primitive guidance improves action generation, we visualize representative policy rollouts with and without PriGo on the same task. Figure 6 shows five representative frames from the "prepare a set of tableware" task, with PriGo-DP in the top row and the baseline diffusion policy, CogACT, in the bottom row. PriGo-DP generates action sequences that are consistent with the primitives required at different stages of the task: rotating and releasing the plate, pushing and releasing the bowl onto the plate, and grasping the cup handle before placing and releasing the cup into the bowl. This primitive-level structure results in a coherent sequence of manipulation behaviors throughout the task.

In contrast, the unguided baseline exhibits several primitive-level errors that accumulate over the rollout. During plate placement, it fails to execute the required rotation, leaving the plate inverted. During bowl placement, it introduces unnecessary rotational motions. Finally, during cup placement, it grasps the inner wall rather than the handle and releases the cup prematurely in mid-air. These examples illustrate how PriGo corrects the local action structure at different stages of a long-horizon task, thereby reducing error accumulation and improving overall execution.

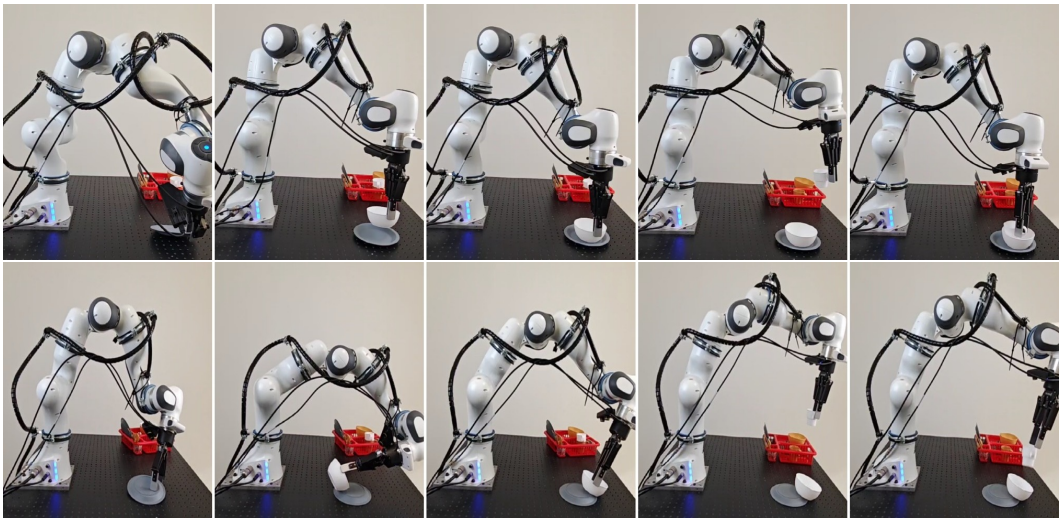


Figure 6: Qualitative comparison between PriGo-DP (top) and the vanilla diffusion policy CogACT (bottom) on the "prepare a set of tableware" task. PriGo produces action sequences that better conform to the required primitive structure across different stages of the task.

Inference efficiency. Finally, we evaluate the computational overhead of PriGo in real-robot deployment through a wall-clock latency analysis against the unguided CogACT baseline. On a single NVIDIA RTX A6000 GPU, CogACT requires approximately 181 ms per inference step, while PriGo requires 208 ms, corresponding to a 15% increase in inference latency. Since each inference cycle predicts a sequence of $k = 16$ actions, PriGo achieves an effective action-generation rate of ~ 77 Hz, which is sufficient for our real-time manipulation setting.

Importantly, the additional per-step inference cost does not translate into longer task execution. PriGo reduces the average task completion time from 2.85 minutes for CogACT to 2.56 minutes, corresponding to an approximately 10% reduction. This result is consistent with the qualitative observations in Figure 6: by producing more structurally coherent action sequences and reducing

primitive-level execution errors, PriGo requires fewer corrective actions and fewer overall steps to complete the task. Thus, the modest inference overhead is offset by more efficient task execution.