Beyond World-Frame Action Heads: Motion-Centric Action Frames for Vision-Language-Action Models

Huoren Yang[1,2] **Jianchao Zhao**[1,2] **Yusong Hu**[2,†] **Qiguan Ou**[2] **Yuyang Gao**[2]
Wei Ke[4] **Yuhang He**[1] **SongLin Dong**[3] **Zhiheng Ma**[3,*] **Yihong Gong**[1,3]
[1]National Key Laboratory of Human-Machine Hybrid Augmented Intelligence,
National Engineering Research Center for Visual Information and Applications,
Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
[2]One Robotics
[3]Faculty of Computility Microelectronics, Shenzhen University of Advanced Technology
[4]School of Software Engineering, Xi'an Jiaotong University
{yanghuoren,zhaojianchao}@stu.xjtu.edu.cn
{wei.ke,ygong}@mail.xjtu.edu.cn heyuhang@xjtu.edu.cn
{dongsl,mazhiheng}@suat-sz.edu.cn ys@onerobot.com
{ou1531037815,hilbertgao}@gmail.com

Abstract

Vision-Language-Action (VLA) models have advanced rapidly with stronger backbones, broader pre-training, and larger demonstration datasets, yet their action heads remain largely homogeneous: most directly predict action commands in a fixed world coordinate frame. We propose **MCF-Proto**, a lightweight action head that equips VLA policies with a Motion-Centric Action Frame (MCF) and a prototype-based action parameterization. At each step, the policy predicts a rotation $R_t \in SO(3)$, composes actions in the transformed local frame from a set of prototypes, and maps them back to the world frame for end-to-end training, using only standard demonstrations without auxiliary supervision. By aligning actions in an observation-conditioned local frame, MCF-Proto concentrates action variation and enables shared prototypes to capture recurring manipulation patterns across tasks. On LIBERO-plus, it achieves the best reported success rate in six of seven perturbation categories, with particularly strong gains under geometric perturbations. Controlled experiments show that the learned frame and prototype composition are complementary, improving mean success from 72.0% to 81.9% over a standard action head. Experiments on LIBERO and two real-world robot platforms further demonstrate its effectiveness for manipulation. These results highlight structured action parameterization as a useful direction for improving VLA policies. Code is available at `https://github.com/Heron0625/MCF-Proto`.

1 Introduction

Vision-Language-Action (VLA) models have become a dominant paradigm for robotic manipulation by combining large-scale vision-language pre-training with imitation learning on broad robot datasets [5, 54, 31, 47, 24, 4]. Recent progress has been driven largely by stronger perceptual backbones, improved language grounding, and more diverse demonstrations [52, 49, 3, 43]. In contrast, the action side of VLA models remains strikingly uniform: most policies still use a standard regression head to predict end-effector commands in a fixed global world coordinate frame [43, 11, 17, 50].

*: Corresponding author. †: Project leader.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

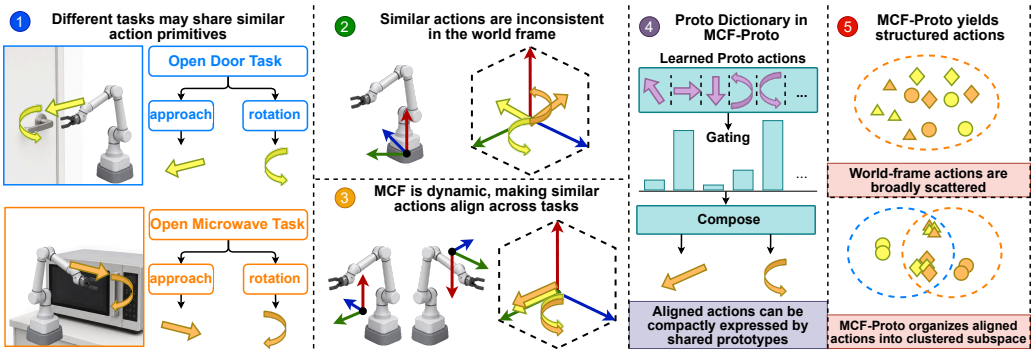

Figure 1: Motivation of MCF-Proto. Similar manipulation behaviors can appear very different when expressed in a fixed world frame, but become better aligned in the learned Motion-Centric Frame (MCF). By combining this local frame with prototype-based action composition, MCF-Proto produces a more compact and structured action representation that is easier to share across tasks and transfer under geometric variation.

While simple and general-purpose, this design does not obviously match the structure of manipulation. Many manipulation behaviors are inherently local and geometric [23, 6, 7, 44, 38]. Actions such as approaching a handle, opening a drawer, inserting along a slot, or rotating around an object-relative axis may share similar interaction structure, yet appear as very different vectors in the world frame. Under such a parameterization, the policy must absorb these variations implicitly in its latent representation while decoding them into a single fixed output space. This motivates a natural question: can VLA policies organize actions more effectively if equipped with a learned local frame and a structured mechanism for action composition within that frame?

To answer this question, we propose **MCF-Proto**, a lightweight action head redesign for standard VLA policies. As shown in Figure 1, instead of directly regressing actions in the world frame, the policy first predicts an observation-conditioned rotation $R_t \in SO(3)$ to define a Motion-Centric Action Frame (MCF). Within this frame, actions are generated as soft combinations of a small shared prototype dictionary, encouraged by orthogonal regularization to remain distinct and non-redundant, and are then mapped back to the world frame for supervision. The entire model is trained end-to-end using only standard demonstrations, without auxiliary labels such as object poses, contact states, or motion directions.

This output-side modification organizes actions into a more concentrated local representation, allowing shared prototypes to capture recurring motion patterns across tasks. MCF-Proto achieves the best reported performance in six of seven LIBERO-plus perturbation categories, including strong gains under changes in camera viewpoint and robot pose. Controlled action-head comparisons show that the learned frame and prototype composition contribute complementary benefits. Results on LIBERO and multi-stage tasks on OpenArm and AgileX Piper further support the effectiveness of the design.

Our contributions are threefold:

- We introduce MCF-Proto, a lightweight VLA action head that combines a Motion-Centric Action Frame, and a shared action prototype dictionary, while requiring only standard action supervision.
- We show that MCF-Proto induces structured local action representations: the learned frame exhibits axes aligned with demonstrated motion, while prototype composition organizes recurring action patterns into a compact, reusable representation.
- We demonstrate that these structural properties improve both task performance and robustness, with particularly strong gains under geometric perturbations on LIBERO-plus.

2 Related works

Behavioral Cloning. Behavioral cloning (BC) has become a strong, scalable paradigm for visuomotor manipulation due to simplicity, stable optimization, and compatibility with large-scale

demonstration data. Recent works show that image-conditioned policies trained with larger datasets, stronger visual encoders, and multi-task formulations can produce increasingly capable manipulation behaviors across diverse settings [22, 42, 43, 11, 50, 19, 9]. These advances have substantially improved perception, policy capacity, and task coverage, establishing BC-style visuomotor policies as competitive alternatives to more complex reinforcement learning pipelines in data-rich regimes [22, 43, 11, 50, 19, 9, 47]. Meanwhile, much of this progress has focused on better observation modeling and larger policy backbones, while the action prediction layer often remains relatively generic, directly regressing end-effector commands in a fixed world or robot frame [5, 31, 11, 17]. Our work is motivated by the view that action parameterization itself is an important design choice: even with a strong backbone, expressing manipulation actions in a more structured local form can make action patterns easier to share across scenes and improve robustness to geometric variation.

Action Frames. A long line of work in robotics has emphasized that manipulation actions are often more naturally expressed in local, object-centric, contact-centric, or task-aligned coordinate systems rather than a single global frame [23, 6, 7, 1, 10]. Such representations can simplify control, improve invariance to changes in scene layout, and better reflect the geometry of physical interaction [29, 7, 44, 14]. Prior approaches have incorporated local frames through object-centric policies, affordance-guided control, equivariant architectures, or explicit geometric reasoning modules [13, 10, 40, 44, 38, 45, 14, 53, 41]. More broadly, these works suggest that choosing an appropriate coordinate system can substantially reduce the burden on downstream control prediction. Our work follows this perspective, but integrates it into a lightweight policy-head design for imitation learning: instead of assuming a predefined object or interaction frame, we predict an observation-conditioned MCF directly from policy features, and use it to express the action before mapping it back to the world frame. In this way, the learned frame serves as a compact intermediate representation for action generation rather than an externally specified geometric input.

Prototype Actions. Our action representation is also related to prior work on motion primitives, dictionary learning, structured latent actions, and compositional policy outputs [21, 32, 28, 16]. Across robotics and machine learning, reusable basis elements or prototype components have been used to capture recurring structure in trajectories, skills, and control policies, often improving parameter efficiency, interpretability, and transfer [21, 32, 37, 35, 18]. Related ideas also appear in mixture-of-experts models, sparse coding, and other compositional prediction frameworks, where outputs are formed from a small set of shared components [39, 12, 26]. Our work adopts this general principle at the action-head level: instead of predicting each action dimension independently, we compose local translational and rotational actions from small sets of shared prototypes, each implemented as a learned linear map to a 3D action increment. The key difference is that this composition is performed in the learned MCF rather than directly in the world frame. This combination allows similar motion patterns to be shared after geometric alignment, leading to a more compact local action representation.

3 Method

We introduce **MCF-Proto**, a lightweight redesign of the action head for vision-language-action policies. Instead of predicting end-effector actions directly in the world frame, MCF-Proto first predicts an observation-conditioned local rotation $R_t \in SO(3)$, uses it to define a Motion-Centric Action Frame (MCF), composes actions in that frame from a small set of shared prototypes, and then maps the result back to the world frame for standard imitation learning.

This design is motivated by two simple observations. First, manipulation actions are often easier to describe in a local geometric frame than in a fixed global one. Second, many manipulation behaviors share reusable local motion patterns even when their world-frame realizations vary across scenes, object poses, or viewpoints. MCF-Proto builds these two biases directly into the policy output layer while leaving the backbone and supervision unchanged.

3.1 Setup

Given an observation o_t, a VLA backbone produces a latent feature $h_t = f_{\text{backbone}}(o_t)$. We consider actions consisting of end-effector translation and rotation increments together with any remaining control dimensions: $a_t^\star = (\Delta x_t^\star, \Delta r_t^\star, a_{t,\text{rest}}^\star)$, where $\Delta x_t^\star \in \mathbb{R}^3$ and $\Delta r_t^\star \in \mathbb{R}^3$. A conventional

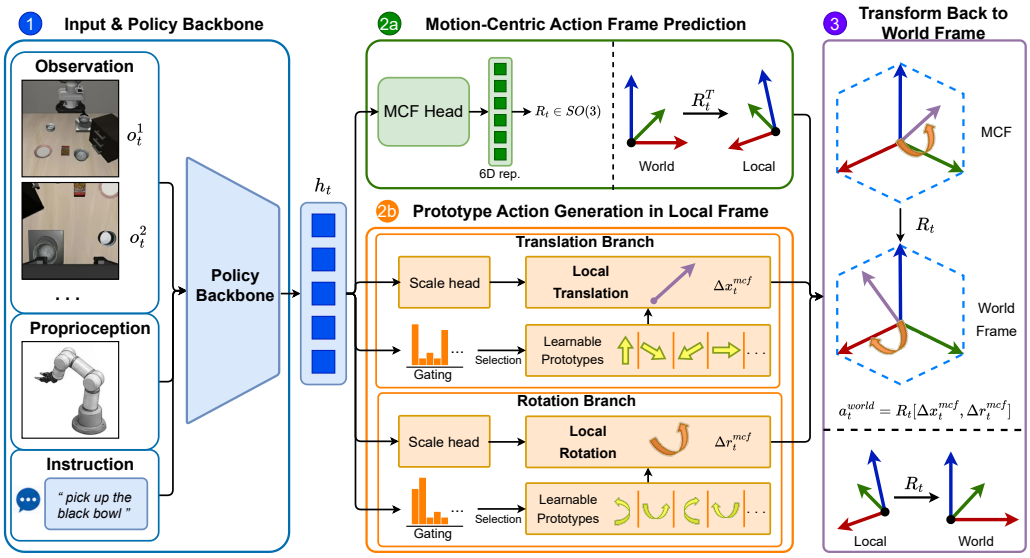

Figure 2: Overview of MCF-Proto. Given observations and task instruction, the policy backbone extracts latent features that are used to predict MCF. Translation and rotation actions are then generated in this local frame through soft combinations of shared prototypes, and finally transformed back to the world frame for standard action supervision.

action head would predict these quantities directly in the world frame. Our goal is to predict the same action target through a structured local representation.

3.2 Motion-Centric Action Frame

As shown in Figure 2, from the policy feature h_t, we predict a rotation $R_t \in SO(3)$ using a lightweight MLP with a continuous 6D rotation parameterization. This rotation defines the local *Motion-Centric Action Frame* at time t.

Let $\Delta x_t^{\text{mcf}} \in \mathbb{R}^3$ and $\Delta r_t^{\text{mcf}} \in \mathbb{R}^3$ denote the translational and rotational increments expressed in the MCF. They are mapped back to the world frame by:

$$\Delta x_t = R_t \Delta x_t^{\text{mcf}}, \qquad \Delta r_t = R_t \Delta r_t^{\text{mcf}}.$$

The final predicted action is $\hat{a}_t = (\Delta x_t, \Delta r_t, \hat{a}_{t,\text{rest}})$, where $\hat{a}_{t,\text{rest}}$ is predicted by a small auxiliary head.

The role of the MCF is to absorb scene-dependent geometric variation before action decoding. The observation-conditioned frame allows similar local manipulation patterns to be shared across states with different world-frame action directions.

3.3 Prototype composition in the local frame

Within the MCF, we parameterize actions using a small set of shared prototypes. Concretely, each prototype is a matrix $B_k \in \mathbb{R}^{3\times d}$ that maps a latent coefficient vector to a 3D local action increment. We maintain separate prototype dictionaries for translation and rotation:

$$B^{\text{trans}} = \{B_k^{\text{trans}}\}_{k=1}^{K_t}, \qquad B^{\text{rot}} = \{B_k^{\text{rot}}\}_{k=1}^{K_r},$$

Given the latent state h_t, the model predicts latent scaling vectors $z^{\text{trans}}, z^{\text{rot}} \in \mathbb{R}^d$, as well as prototype gating logits for translation and rotation. After a softmax, these produce gating distributions π_t^x and π_t^r, respectively. The local action is then composed as:

$$\Delta x_t^{\text{mcf}} = \sum_{k=1}^{K_t} \pi_t^x[k]\, B_k^{\text{trans}} z^{\text{trans}}, \qquad \Delta r_t^{\text{mcf}} = \sum_{k=1}^{K_r} \pi_t^r[k]\, B_k^{\text{rot}} z^{\text{rot}}.$$

This local prototype parameterization encourages the policy to express actions through a small set of shared prototype mappings modulated by latent scale coefficients. Combined with the learned local frame, it allows similar manipulation behaviors to exhibit shared structure even when their world-frame directions differ. In practice, this leads to a substantially more concentrated action distribution in the learned local frame than in the original world frame.

The two components of MCF-Proto play distinct but complementary roles. The predicted local frame reduces scene-dependent geometric variation by aligning actions into an observation-conditioned coordinate system, while the prototype dictionaries exploit the resulting alignment to capture recurrent motion patterns through a shared low-dimensional parameterization. Without the local frame, similar behaviors may correspond to very different world-frame directions and are therefore harder to organize consistently across tasks. Without prototype composition, the local frame alone provides geometric alignment but does not explicitly encourage shared structure or low-dimensional organization in the action output space. Their combination yields a more structured representation in which directional alignment and prototype-based composition emerge jointly.

To keep the prototype dictionaries well-conditioned, we apply an orthogonality regularizer:

$$\mathcal{L}_{\text{ortho}} = \left\|(B^{\text{trans}})^\top B^{\text{trans}} - I\right\|_F^2 + \left\|(B^{\text{rot}})^\top B^{\text{rot}} - I\right\|_F^2 .$$

3.4 Learning objective

MCF-Proto is trained end-to-end by supervising the final predicted end-effector incremental action in the world frame. Let $a_t^\star$ denote the demonstrated action and let $\hat{a}_t$ denote the predicted action. We supervise translational and gripper dimensions using an L_1 loss, and rotational dimensions using a Smooth L_1 loss in radians. Let $I_{\text{trans}}, I_{\text{rot}}, I_{\text{grip}}$ denote the translation, rotation, and gripper index sets in the full action vector. The action supervision term is:

$$\mathcal{L}_{\text{act}} = \|\hat{a}_t[I_{\text{trans}}] - a_t^\star[I_{\text{trans}}]\|_1 + \|\hat{a}_t[I_{\text{grip}}] - a_t^\star[I_{\text{grip}}]\|_1 + \text{SmoothL1}(\hat{a}_t[I_{\text{rot}}], a_t^\star[I_{\text{rot}}]) .$$

To encourage temporal consistency of the learned motion-centric frame, we regularize consecutive frame rotations to vary smoothly over time. Intuitively, the local frame is encouraged to evolve smoothly over time rather than rotating abruptly to follow every instantaneous action change. When the action direction shifts across task stages, the model can explain such variation through different axes within the frame, while preserving a relatively stable local basis. Let R_t denote the frame rotation at time t. We define $\Delta R_t = R_t R_{t-1}^\top, \cos\theta_t = \frac{\text{tr}(\Delta R_t)-1}{2}$, and use the smoothness loss $\mathcal{L}_{\text{smooth}} = 1 - \cos\theta_t$. The final objective is:

$$\mathcal{L} = \mathcal{L}_{\text{act}} + \lambda_{\text{ortho}}\mathcal{L}_{\text{ortho}} + \lambda_{\text{smooth}}\mathcal{L}_{\text{smooth}}.$$

Implementation details. The translation and rotation dictionaries are learnable parameters shared across tasks and time steps within each training run. They are randomly initialized, column-normalized, and optimized end-to-end. We use 16 prototypes for each dictionary and latent dimension 3, with regularizer weights 10^{-2} for orthogonality and 10^{-3} for temporal smoothness. Training and evaluation details are provided in the appendix.

4 Experiments

We evaluate MCF-Proto from four perspectives: benchmark performance, robustness under distribution shift, analysis of the learned local action representation, and real-world validation. We first report results on LIBERO [27] and LIBERO-plus [15], then analyze how the learned motion-centric frame reshapes the action space through concentration and geometric compatibility diagnostics, and finally present real-world results and hierarchical ablations.

4.1 Benchmarks and metrics

LIBERO. We evaluate on the standard LIBERO benchmark, including the Spatial, Object, Goal, and Long suites. We report task success rate (%) on each suite and the average across all suites.
LIBERO-plus. We further evaluate robustness on LIBERO-plus, which introduces seven perturbation categories: Camera, Robot, Language, Light, Background, Noise, and Layout. We report success rate (%) for each category.

4.2 Baselines

We compare against a diverse set of recent VLA and robot policy baselines. On LIBERO, we include π_0+FAST [33], OpenVLA-OFT [25], π_0 [2], FLOWER [36], GR00T-N1.5 [30], and BEAST [51]. On LIBERO-plus, we compare against OpenVLA [24], OpenVLA-OFT, π_0, π_0-fast, Nora [20], WorldVLA [8], UniVLA [48], and RIPT-VLA [46].

Table 1: Experimental Results on the LIBERO Benchmark. Success rate (%) is reported for each task suite. **Bold** indicates the best performance, underlined indicates the second best.

Model	Spatial	Object	Goal	Long	Avg
π_0+FAST	96.4	96.8	88.6	60.2	85.5
OpenVLA-OFT	97.6	98.4	97.9	94.5	97.1
π_0	96.8	98.8	95.8	85.2	94.1
FLOWER	97.1	96.7	95.6	93.5	95.7
GR00T-N1.5	92.0	92.0	86.0	76.0	86.5
BEAST	92.9	97.5	93.1	86.4	92.5
Ours	**98.6±0.2**	**98.8±0.2**	**98.4±0.2**	**95.0±0.5**	**97.7±0.3**

Table 2: LIBERO-plus success rates (%). Our results are means ± standard deviations across three independently trained seeds, each evaluated with one rollout per instance over the full 10,030-instance benchmark. External baseline results are taken from prior reports and use their respective backbones and training procedures. Best is bold; second best is underlined.

Model	Camera	Robot	Language	Light	Background	Noise	Layout
OpenVLA	1.1	4.1	26.8	4.4	25.3	19.3	31.6
OpenVLA-OFT	59.7	37.2	**81.5**	85.8	92.4	76.7	77.1
π_0	15.8	6.6	61.0	79.6	78.5	79.4	70.4
π_0-fast	66.4	24.8	63.3	73.0	67.7	75.8	70.3
Nora	4.0	41.1	67.0	31.0	50.5	17.6	63.9
WorldVLA	0.3	30.2	44.2	29.4	14.5	12.2	39.4
UniVLA	4.3	50.3	71.8	59.1	80.0	25.3	34.3
RIPT-VLA	58.3	36.7	80.1	87.9	90.4	73.8	76.5
Ours	**69.7±0.2**	**66.0±0.2**	80.1±0.3	**98.3±0.1**	**93.3±0.2**	**84.0±0.3**	**82.1±0.2**

4.3 Main results on LIBERO

Table 1 reports LIBERO success rates. MCF-Proto achieves the highest average score of 97.7%, compared with 97.1% for OpenVLA-OFT, with improvements on Spatial, Goal, and Long. These results show that the structured action head maintains strong in-distribution performance; LIBERO-plus evaluates its robustness under distribution shift.

Tables 1 and 2 compare complete systems. To isolate the contribution of action parameterization, our ablations use the same backbone, data, training schedule, and approximately matched head parameters.

4.4 Main results on LIBERO-plus

Table 2 reports the results on LIBERO-plus. Our method performs best on six of the seven perturbation categories: Camera (69.7%), Robot (66.0%), Light (98.3%), Background (93.3%), Noise (84.0%), and Layout (82.1%). On Language, it achieves 80.1%, staying close to the best reported baseline (81.5%).

The component ablations in Table 3 reveal complementary roles for geometric alignment and prototype composition. Relative to the MLP baseline, MCF-only improves Camera by 8.7 percentage points and Robot by 12.6 points, while World-Proto improves Light by 7.6 points, Background by 5.4 points, and Noise by 6.3 points. Combining the learned frame with prototypes outperforms either component alone in every category, raising mean success to 80.5%; regularization further

Table 3: Controlled action-head ablation on LIBERO-plus. All variants use the same backbone, data, optimization schedule, and approximately matched head parameter counts.

Variant	Camera	Robot	Language	Light	Backgr.	Noise	Layout	Mean
BC w/ matched MLP	58.5	48.9	79.3	88.0	85.4	74.6	69.3	72.0
MCF-only	67.2	61.5	79.6	89.8	87.0	76.3	74.1	76.5
World-Proto	60.3	51.8	79.5	95.6	90.8	80.9	69.6	75.5
MCF + Proto, no reg.	68.9	64.8	79.9	97.5	92.4	82.7	77.3	80.5
MCF-Proto (full)	**69.7**	**66.0**	**80.1**	**98.3**	**93.3**	**84.0**	**82.1**	**81.9**

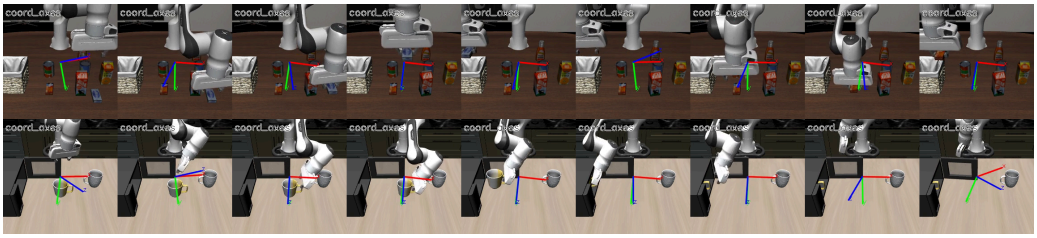

Figure 3: Learned MCF visualization on LIBERO. The predicted MCF provides a smooth local basis whose axes remain strongly compatible with the dominant motion direction in two representative manipulation tasks, despite being learned without explicit directional supervision.

improves it to 81.9%. Language success remains similar across variants (79.3–80.1%). Together, these comparisons show how local geometric alignment and shared action composition contribute to robustness under different types of perturbation.

4.5 Local-frame action concentration analysis

Table 4: Quantitative comparison of action concentration metrics between world-frame and MCF

Metric	World (mean±std)	MCF (mean±std)	Relative Change
Covariance Trace	6.71 ± 1.93	5.55 ± 2.25	-17.8%
Average Pairwise Distance	3.41 ± 0.46	2.93 ± 0.53	-14.0%
PCA Top-3 Explained Variance	0.77 ± 0.03	0.90 ± 0.03	+16.7%
Effective Rank	4.03 ± 0.54	1.94 ± 0.21	-51.3%

To quantify how the learned local frame changes the structure of the action space, we compare world-frame actions and local-frame actions using four task-wise statistics: Covariance Trace, Average Pairwise Distance, Top-3 PCA Explained Variance, and Effective Rank. As shown in Table 4, the local-frame representation is consistently more concentrated than the world-frame representation across all four metrics. Specifically, local-frame actions have lower covariance trace (5.55 vs. 6.71) and lower average pairwise distance (2.93 vs. 3.41), while showing higher Top-3 PCA explained variance (0.90 vs. 0.77) and lower effective rank (1.94 vs. 4.03).

These results indicate that the learned motion-centric frame reorganizes action variability into a lower-dimensional and more concentrated local representation. Figure 4 illustrates the same effect qualitatively: in the local frame, task-specific groupings become more distinguishable than in the world frame, suggesting that geometric alignment makes recurring manipulation patterns easier to separate and reuse.

This concentration effect is important because it provides a quantitative explanation for why prototype-based composition becomes more effective in the learned local frame. When action variation is distributed across fewer dominant directions, a small shared prototype dictionary can capture more of the task-relevant structure with less redundancy. In this sense, the MCF does not merely rotate the action space; it reorganizes it into a form that is more amenable to compact reuse.

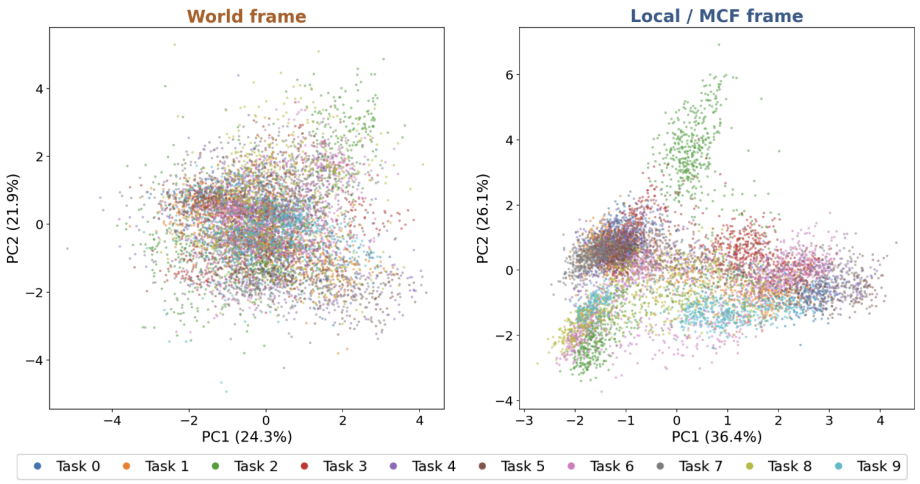

Figure 4: Visualization of action distributions in the world frame and the learned local frame. Compared with world-frame actions, actions expressed in the Motion-Centric Frame are more concentrated and form more separable task-specific clusters, suggesting that the learned local representation better captures shared and reusable manipulation structure.

4.6 Alignment with demonstrated motion

Table 5: Angular error (°) between MCF axes and demonstrated end-effector motion directions

Metric	**place**	**door-close**	**knob-turn**	**drawer-close**	**insert**
MCF only	$35.4^\circ \pm 3.1^\circ$	$34.7^\circ \pm 1.4^\circ$	$35.7^\circ \pm 2.9^\circ$	$35.4^\circ \pm 7.5^\circ$	$32.8^\circ \pm 5.9^\circ$
MCF-Proto	$18.8^\circ \pm 3.9^\circ$	$18.2^\circ \pm 1.5^\circ$	$10.8^\circ \pm 3.4^\circ$	$17.4^\circ \pm 2.7^\circ$	$13.2^\circ \pm 6.8^\circ$

We further assess whether the learned motion-centric frame captures useful geometric structure by measuring how well its axes align with demonstrated local motion on representative LIBERO-Long tasks. For each time step, we compute the minimum unsigned angle between the demonstrated end-effector displacement direction and the three axes of the predicted frame. A smaller angle indicates that at least one learned axis is compatible with the demonstrated local motion.

Table 5 shows that MCF-Proto reduces the minimum axis–motion angle from approximately 33–36 degrees for MCF-only to 11–19 degrees. The consistent reduction across tasks shows that prototype composition encourages the frame axes to align with demonstrated motion, complementing the concentration results above. Additional reference-frame comparisons are provided in Table **??** in the appendix.

4.7 Real-world experiments on OpenArm

We compare with π_0 [2] and $\pi_{0.5}$ [34].

Table 6: Real-world evaluation on OpenArm. Task 1 reports the success rate (%) of placing at least the first $k \in \{1, 2, 3, 4\}$ test tubes correctly, together with the average number of completed tubes (out of four). Tasks 2 and 3 report overall task success rate. Higher is better for all metrics.

	1: Test-tube placement					**2: Bi. transfer**	**3: Bowl stack**
Method	**@1**	**@2**	**@3**	**@4**	**Count (/4)**	**Success**	**Success**
π_0	64.0	26.0	14.0	8.0	1.12	32.0	66.0
$\pi_{0.5}$	80.0	32.0	24.0	18.0	1.54	40.0	72.0
Ours	**84.0**	**56.0**	**42.0**	**38.0**	**2.20**	**64.0**	**82.0**

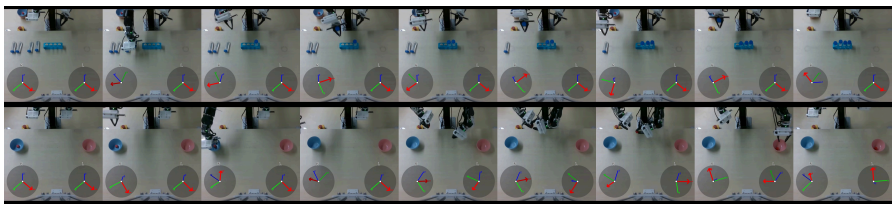

Figure 5: The predicted local frame maintains a stable local basis whose axes remain geometrically compatible with the principal task motion direction in real-world rollouts, suggesting that similar geometric structure also emerges in real-world manipulation.

We evaluate MCF-Proto on OpenArm[1] using three manipulation tasks: placing four test tubes onto a rack, bimanually transferring a block between bowls, and stacking two bowls. Table 6 reports task success and the number of completed tube placements. We additionally evaluate the same tasks on a mechanically distinct AgileX Piper platform, with results provided in the appendix. Across both platforms, MCF-Proto improves multi-stage task completion.

These results suggest that the benefits of local action alignment and prototype-based composition also extend to real-world manipulation. The advantage is especially meaningful in multi-stage tasks, where small low-level prediction errors can accumulate over time and destabilize the full trajectory.

4.8 Ablation studies

Table 7: Controlled frame comparison with the same backbone and training recipe. Frame-based variants use prototype composition. Means are success rates (%).

Action head	LIBERO	LIBERO-plus
Parameter-matched MLP	93.8	72.0
World-Proto	94.4	75.5
End-effector frame + Proto	95.6	77.5
Offline per-task PCA frame + Proto	96.4	79.0
MCF-Proto	97.7	81.9

Table 7 compares frame choices under the same backbone and training budget. MCF-Proto achieves 81.9% mean success on LIBERO-plus, compared with 75.5% for World-Proto, 77.5% for an end-effector frame, and 79.0% for a per-task PCA frame. The 2.9-point improvement over the PCA frame demonstrates the benefit of adapting the action frame to the current observation beyond alignment with fixed offline task statistics.

Table 8: Hierarchical ablation of MCF-Proto. Avg denotes the average task success rate over all evaluation suites.

Method	**MCF**	**Proto**	L_{ortho}	L_{smooth}	**Avg (%)**
BC w/ MLP head	✗	✗	✗	✗	93.8±0.6
MCF only	✓	✗	✗	✗	94.7±0.5
World-Proto	✗	✓	✗	✗	94.4±0.6
MCF + Proto	✓	✓	✗	✗	96.9±0.3
MCF + Proto + L_{ortho}	✓	✓	✓	✗	97.4±0.2
MCF-Proto (full)	✓	✓	✓	✓	97.7±0.3

Table 8 shows hierarchical ablations of MCF-Proto. Introducing the motion-centric frame alone improves over the BC baseline ($93.8\% \rightarrow 94.7\%$), and prototype composition alone in the world frame is also beneficial (94.4%). Combining both yields a substantially larger gain (96.9%), indicating that MCF and prototype composition are complementary. Adding $\mathcal{L}_{\text{ortho}}$ further improves performance

[1] `https://openarm.dev`

to 97.4%, and adding the temporal smoothness regularizer $\mathcal{L}_{\text{smooth}}$ gives the full model with the best average success rate (97.7%).

Together, the representation analyses and ablations support a mutually reinforcing relationship: the learned frame aligns actions across states and tasks for prototype reuse, while prototype-based prediction encourages the frame to align with demonstrated motion. The combined design yields larger performance gains than either component alone.

5 Discussion and Limitations

MCF-Proto shows that action parameterization is an important but underexplored design axis for vision-language-action policies. By redesigning only the action head, the method consistently improves LIBERO performance and LIBERO-plus robustness, with especially clear gains under camera and robot perturbations. Our analyses further suggest that these gains stem from a more structured action representation: the learned motion-centric frame concentrates action variation in a local coordinate system, while prototype-based composition yields more compact and reusable motion structure after alignment. These results indicate that lightweight geometric and compositional structure on the output side can provide a useful inductive bias for manipulation learning.

Our evaluation covers LIBERO, LIBERO-plus, and a limited set of real-world manipulation tasks. Broader validation is needed for dynamic, deformable, and contact-rich settings. The learned frame is an implicit representation; its axis labels depend on the coordinate convention, as discussed in the appendix. As a behavioral cloning method, MCF-Proto also depends on the coverage and quality of demonstrations. Future work will explore the same action parameterization at larger scale and with online adaptation.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants 62576224, 62606333, U21B2048, 62302382, and 62376209; in part by the Guangdong Provincial Key Laboratory of Computility Microelectronics under Grant 2024B1212010007; in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2026A1515011733; in part by the Characteristic Innovation Project of Ordinary Universities in Guangdong Province under Grant 2024KTSCX026; in part by the Shenzhen Key Technical Projects under Grant CJGJZD20220517141605013; in part by the China Postdoctoral Science Foundation under Grant 2024M752584; and in part by the Research and Development Fund Project of China Huaneng Group Co., Ltd. under Grant HNKJ24-H144.

References

[1] Tohid Alizadeh, Sylvain Calinon, and Darwin G Caldwell. Learning from demonstrations with partially observable task parameters. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3309–3314. IEEE, 2014.

[2] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2026. URL `https://arxiv.org/abs/2410.24164`.

[3] Rogerio Bonatti, Sai Vemprala, Shuang Ma, Felipe Frujeri, Shuhang Chen, and Ashish Kapoor. Pact: Perception-action causal transformer for autoregressive robotics pre-training. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3621–3627. IEEE, 2023.

[4] Konstantinos Bousmalis, Giulia Vezzani, Dushyant Rao, Coline Devin, Alex X Lee, Maria Bauzá, Todor Davchev, Yuxiang Zhou, Agrim Gupta, Akhil Raju, et al. Robocat: A self-improving generalist agent for robotic manipulation. *arXiv preprint arXiv:2306.11706*, 2023.

[5] 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.

[6] Sylvain Calinon. A tutorial on task-parameterized movement learning and retrieval. *Intelligent service robotics*, 9(1):1–29, 2016.

[7] Sylvain Calinon, Danilo Bruno, and Darwin G Caldwell. A task-parameterized probabilistic model with minimal intervention control. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3339–3344. IEEE, 2014.

[8] Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, Deli Zhao, and Hao Chen. Worldvla: Towards autoregressive action world model, 2025. URL `https://arxiv.org/abs/2506.21539`.

[9] Shizhe Chen, Ricardo Garcia, Cordelia Schmid, and Ivan Laptev. Polarnet: 3d point clouds for language-guided robotic manipulation. *arXiv preprint arXiv:2309.15596*, 2023.

[10] Xiaotong Chen, Kaizhi Zheng, Zhen Zeng, Cameron Kisailus, Shreshtha Basu, James Cooney, Jana Pavlasek, and Odest Chadwicke Jenkins. Manipulation-oriented object perception in clutter through affordance coordinate frames. In *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*, pages 186–193. IEEE, 2022.

[11] 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*, 44(10-11):1684–1704, 2025.

[12] Zichen Jeff Cui, Yibin Wang, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. From play to policy: Conditional behavior generation from uncurated robot data. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=c7rM7F7jQjN`.

[13] Coline Devin, Pieter Abbeel, Trevor Darrell, and Sergey Levine. Deep object-centric representations for generalizable robot learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7111–7118. IEEE, 2018.

[14] Ben Eisner, Yi Yang, Todor Davchev, Mel Vecerik, Jonathan Scholz, and David Held. Deep se(3)-equivariant geometric reasoning for precise placement tasks. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Learning Representations*, volume 2024, pages 26866–26886, 2024. URL `https://proceedings.iclr.cc/paper_files/paper/2024/file/71dbba3a3731f8eb9dcfa682460627a3-Paper-Conference.pdf`.

[15] Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

[16] Javier Felip, Janne Laaksonen, Antonio Morales, and Ville Kyrki. Manipulation primitives: A paradigm for abstraction and execution of grasping and manipulation tasks. *Robotics and Autonomous Systems*, 61 (3):283–296, 2013.

[17] Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Conference on robot learning*, pages 158–168. PMLR, 2022.

[18] Felix Frank, Alexandros Paraschos, Patrick van der Smagt, and Botond Cseke. Constrained probabilistic movement primitives for robot trajectory adaptation. *IEEE Transactions on Robotics*, 38(4):2276–2294, 2021.

[19] Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. Rvt: Robotic view transformer for 3d object manipulation. In *Conference on Robot Learning*, pages 694–710. PMLR, 2023.

[20] Chia-Yu Hung, Qi Sun, Pengfei Hong, Amir Zadeh, Chuan Li, U-Xuan Tan, Navonil Majumder, and Soujanya Poria. Nora: A small open-sourced generalist vision language action model for embodied tasks, 2025. URL `https://arxiv.org/abs/2504.19854`.

[21] Auke Jan Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2):328–373, 2013.

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

[23] Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.

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

[25] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

[26] Seungjae Lee, Yibin Wang, Haritheja Etukuru, H Jin Kim, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Behavior generation with latent actions. *arXiv preprint arXiv:2403.03181*, 2024.

[27] 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.

[28] J Daniel Morrow and Pradeep K Khosla. Manipulation task primitives for composing robot skills. In *Proceedings of International Conference on Robotics and Automation*, volume 4, pages 3354–3359. IEEE, 1997.

[29] Jun Nakanishi, Rick Cory, Michael Mistry, Jan Peters, and Stefan Schaal. Operational space control: A theoretical and empirical comparison. *The International Journal of Robotics Research*, 27(6):737–757, 2008.

[30] NVIDIA, Johan Bjorck, Nikita Cherniadev Fernando Castañeda, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots. In *ArXiv Preprint*, March 2025.

[31] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[32] Alexandros Paraschos, Christian Daniel, Jan R Peters, and Gerhard Neumann. Probabilistic movement primitives. *Advances in neural information processing systems*, 26, 2013.

[33] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models, 2025. URL `https://arxiv.org/abs/2501.09747`.

[34] Physical Intelligence, Kevin Black, Noah Brown, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. URL `https://arxiv.org/abs/2504.16054`.

[35] Michael Przystupa, Faezeh Haghverd, Martin Jagersand, and Samuele Tosatto. Deep probabilistic movement primitives with a bayesian aggregator. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3704–3711. IEEE, 2023.

[36] Moritz Reuss, Hongyi Zhou, Marcel Rühle, Ömer Erdinç Yağmurlu, Fabian Otto, and Rudolf Lioutikov. FLOWER: Democratizing generalist robot policies with efficient vision-language-flow models. In *9th Annual Conference on Robot Learning*, 2025. URL `https://openreview.net/forum?id=JeppaebLRD`.

[37] Elmar Rückert and Andrea d'Avella. Learned parametrized dynamic movement primitives with shared synergies for controlling robotic and musculoskeletal systems. *Frontiers in computational neuroscience*, 7: 138, 2013.

[38] Hyunwoo Ryu, Hong-in Lee, Jeong-Hoon Lee, and Jongeun Choi. Equivariant descriptor fields: Se (3)-equivariant energy-based models for end-to-end visual robotic manipulation learning. *arXiv preprint arXiv:2206.08321*, 2022.

[39] Nur Muhammad Shafiullah, Zichen Cui, Ariuntuya Arty Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone. *Advances in neural information processing systems*, 35:22955–22968, 2022.

[40] Mohit Sharma, Jacky Liang, Jialiang Zhao, Alex LaGrassa, and Oliver Kroemer. Learning to compose hierarchical object-centric controllers for robotic manipulation. *arXiv preprint arXiv:2011.04627*, 2020.

[41] William Shen, Ge Yang, Alan Yu, Jansen Wong, Leslie Pack Kaelbling, and Phillip Isola. Distilled feature fields enable few-shot language-guided manipulation. *arXiv preprint arXiv:2308.07931*, 2023.

[42] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on robot learning*, pages 894–906. PMLR, 2022.

[43] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.

[44] Anthony Simeonov, Yilun Du, Andrea Tagliasacchi, Joshua B Tenenbaum, Alberto Rodriguez, Pulkit Agrawal, and Vincent Sitzmann. Neural descriptor fields: Se (3)-equivariant object representations for manipulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6394–6400. IEEE, 2022.

[45] Anthony Simeonov, Yilun Du, Yen-Chen Lin, Alberto Rodriguez Garcia, Leslie Pack Kaelbling, Tomás Lozano-Pérez, and Pulkit Agrawal. Se (3)-equivariant relational rearrangement with neural descriptor fields. In *Conference on Robot Learning*, pages 835–846. PMLR, 2023.

[46] Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive post-training for vision-language-action models, 2025. URL `https://arxiv.org/abs/2505.17016`.

[47] 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.

[48] Yuqi Wang, Xinghang Li, Wenxuan Wang, Junbo Zhang, Yingyan Li, Yuntao Chen, Xinlong Wang, and Zhaoxiang Zhang. Unified vision-language-action model, 2025. URL `https://arxiv.org/abs/2506.19850`.

[49] 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.

[50] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[51] Hongyi Zhou, Weiran Liao, Xi Huang, Yucheng Tang, Fabian Otto, Xiaogang Jia, Xinkai Jiang, Simon Hilber, Ge Li, Qian Wang, Ömer Erdinç Yağmurlu, Nils Blank, Moritz Reuss, and Rudolf Lioutikov. Beast: Efficient tokenization of b-splines encoded action sequences for imitation learning, 2025. URL `https://arxiv.org/abs/2506.06072`.

[52] Y Zhu et al. Vima: General robot manipulation with multimodal prompts. In *International Conference on Learning Representations (ICLR)*, 2023.

[53] Yifeng Zhu, Zhenyu Jiang, Peter Stone, and Yuke Zhu. Learning generalizable manipulation policies with object-centric 3d representations. *arXiv preprint arXiv:2310.14386*, 2023.

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

Code Repository

The implementation and training instructions are available at:

`https://github.com/Heron0625/MCF-Proto`

A Why compact local representations favor directional alignment

Across three independently trained models, the measured frame agreement after sign/permutation matching is $4.2^\circ \pm 0.8^\circ$. This is an empirical cross-seed diagnostic; sign/permutation matching accounts for a discrete subset of the frame–dictionary coordinate freedom.

In the main paper, MCF-Proto combines a learned local action frame with prototype-based action composition. Empirically, this joint design produces two closely related effects: actions become more concentrated when expressed in the learned local frame, and the frame itself develops stable axes that align strongly with demonstrated end-effector motion.

This appendix provides a simple theoretical view of that connection. We study an idealized setting in which the learning system prefers local coordinates that yield a more compact action representation. Under standard symmetry assumptions, the preferred frame aligns its axes with the principal directions of the action distribution. This is a design-motivation analysis rather than an exact characterization of the trained MCF-Proto objective: the actual reconstruction-plus-regularization loss has a joint frame/prototype gauge freedom, so compactness and axis alignment should be interpreted as empirically emerging structure rather than uniquely identified consequences of the loss.

Let $a \in \mathbb{R}^d$ be a zero-mean action in the world frame, and let

$$u = R^\top a$$

denote its representation in a learned local frame $R \in SO(d)$. Since reconstruction alone is invariant to orthogonal changes of basis, any preference over R must come from additional structure imposed on the local coordinates u. Here we model that preference through the objective

$$\min_{R \in SO(d)} J(R), \qquad J(R) := \mathbb{E}\|R^\top a\|_1,$$

which favors frames in which action variation is more concentrated across coordinates.

Assume a follows an elliptically symmetric distribution with covariance

$$\Sigma = \mathbb{E}[aa^\top].$$

Then a can be written as

$$a = \Sigma^{1/2} z,$$

where z is a spherically symmetric random vector with identity covariance.

Proposition 1 *Under the above assumptions,*

$$\mathcal{J}(R) = c \sum_{i=1}^{d} \sqrt{(R^\top \Sigma R)_{ii}}$$

for some constant $c = \mathbb{E}|z_1| > 0$. Moreover, $\mathcal{J}(R)$ is minimized when $R^\top \Sigma R$ is diagonal. Equivalently, the minimizing frame R has columns given by eigenvectors of Σ. Thus, up to permutation and sign flips, the learned coordinate axes align with the principal directions of the training action distribution.

Proof 1 *Let r_i denote the i-th column of R. Then the i-th local coordinate is*

$$u_i = e_i^\top R^\top a = r_i^\top a.$$

Since $a = \Sigma^{1/2} z$,

$$u_i = r_i^\top \Sigma^{1/2} z.$$

By spherical symmetry of z, the distribution of $r_i^\top \Sigma^{1/2} z$ depends only on its standard deviation. Hence

$$\mathbb{E}|u_i| = c\sqrt{\mathrm{Var}(u_i)} = c\sqrt{r_i^\top \Sigma r_i} = c\sqrt{(R^\top \Sigma R)_{ii}},$$

where $c = \mathbb{E}|z_1| > 0$ is independent of R. Summing over coordinates gives

$$\mathcal{J}(R) = \sum_{i=1}^{d} \mathbb{E}|u_i| = c \sum_{i=1}^{d} \sqrt{(R^\top \Sigma R)_{ii}}.$$

Now let $d(R)$ denote the diagonal vector of $R^\top \Sigma R$. By the Schur–Horn theorem, $d(R)$ is majorized by the eigenvalue vector $\lambda(\Sigma)$. Since the function $x \mapsto \sqrt{x}$ is concave on $\mathbb{R}_+$, Karamata's inequality implies

$$\sum_{i=1}^{d} \sqrt{(R^\top \Sigma R)_{ii}} \geq \sum_{i=1}^{d} \sqrt{\lambda_i},$$

with equality when $R^\top \Sigma R$ is diagonal.

Finally, if $R^\top \Sigma R = \Lambda$ for some diagonal matrix Λ, then

$$\Sigma R = R\Lambda.$$

Therefore, each column r_i of R satisfies

$$\Sigma r_i = \Lambda_{ii} r_i,$$

so r_i is an eigenvector of Σ. Hence the minimizing frame aligns its axes with the eigen-directions, i.e., the principal directions, of the training action distribution. The solution is unique only up to permutation of axes and sign flips, and is additionally non-unique within eigenspaces corresponding to repeated eigenvalues.

B Experimental Details

Controlled action-head comparisons use Qwen3-VL-4B-Instruct with the same demonstrations and optimization schedule for all variants. The MLP and MCF-Proto heads contain approximately 42.0M and 42.2M parameters, respectively. The shared training recipe is specified below. We evaluate on LIBERO (Spatial, Object, Goal, and Long) and LIBERO-plus. Training uses the full LIBERO mixture without a held-out validation split. Input images are resized to 224×224, and proprioceptive inputs use 6 active state dimensions from the 8-dimensional robot state. LIBERO uses 50 rollouts per task. Each LIBERO-plus evaluation pass uses one rollout per task instance over 10,030 instances covering seven perturbation categories. The LIBERO-plus main-table uncertainties are standard deviations of per-category success rates across three independently trained models, each evaluated on the complete instance set; they are not standard errors or per-instance rollout variances.

MCF-Proto uses Qwen3-VL-4B-Instruct as the vision-language backbone and predicts 8-action chunks. Each action has 7 dimensions. We use latent dimension $d = 3$, with $K_t = 16$ translation prototypes and $K_r = 16$ rotation prototypes.

Training uses AdamW with cosine decay and 5k-step warmup. The base learning rate is 2.5×10^{-5}. Models are trained for 40k steps on 4 NVIDIA A-100 GPUs with effective batch size 128. The total training time cost about 36 hours. At inference, we use the 40k-step checkpoint and execute chunked action prediction with chunk size 8.

C Prototype Usage Analysis

To understand how the learned action vocabulary is utilized across tasks, we analyze the prototype gating distribution π of the trained model on the LIBERO benchmark. For each task, we collect per-frame gating weights from all episodes in the training split and compute the mean π over frames, yielding a tasks × K matrix for both translation and rotation prototypes (K = 16). The resulting heatmaps are shown in Figure 6. Two complementary patterns are visible. First, a set of prototypes receives consistently high activation across all ten tasks, forming a shared backbone of task-agnostic motion primitives. These broadly-used prototypes correspond to elementary manipulation

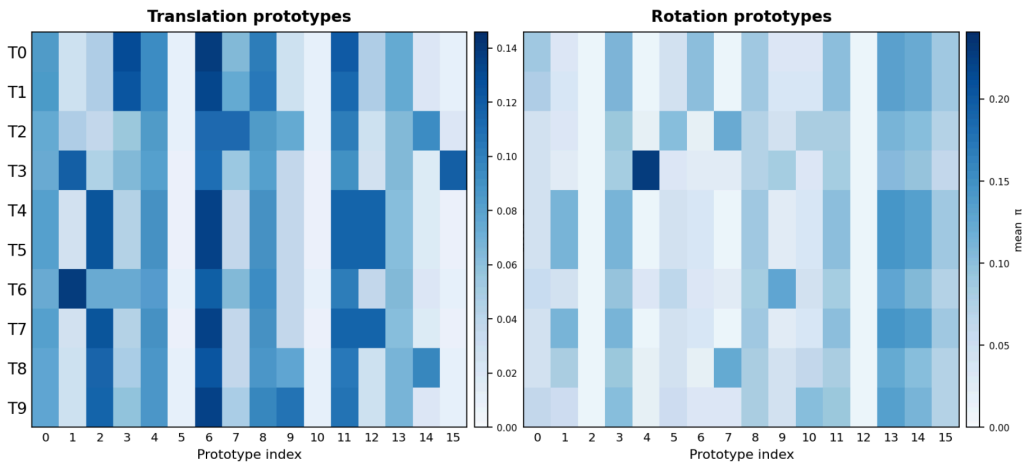

Figure 6: Prototype usage distribution of the learned action model across LIBERO-long tasks (K = 16)

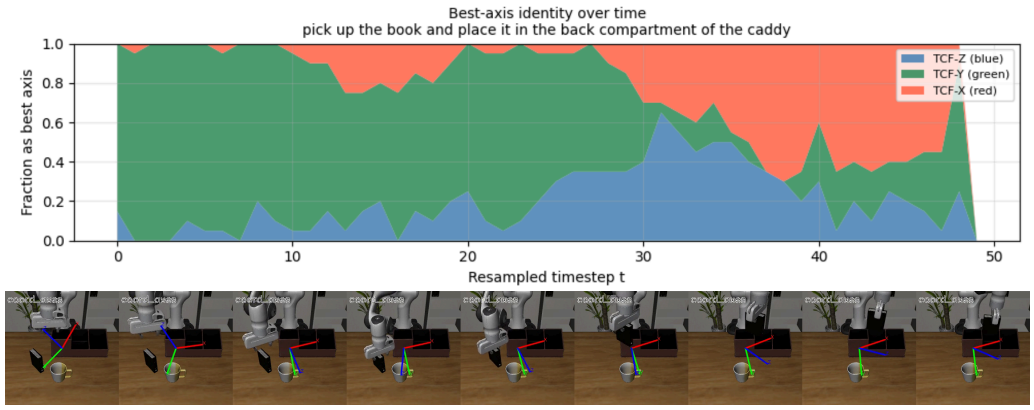

Figure 7: Dominant MCF axis over time for task "pick up the book and place it in the black compartment of the caddy".

phases—reaching toward an object, grasping, lifting, and returning—that appear in virtually every task regardless of scene geometry or object type. Second, a distinct subset of prototypes exhibits strong task-group specificity. In the translation panel, tasks T4, T5, and T7 share elevated weights on a common prototype associated with downward placement trajectories, while T6 and T3 preferentially activate a separate prototype reflecting approach motions toward the stove region. The rotation panel reveals an even sharper specialization: T3 (turn knob) concentrates nearly a quarter of its rotational gating mass onto a single prototype, reflecting the large, structured wrist rotation unique to that task—a pattern absent in all other tasks. Taken together, these results support the hypothesis that the MCF-Proto action head learns a compositional action vocabulary: a compact set of reusable primitives covers universal motion structure, while a sparse collection of specialized prototypes captures task-conditioned variation. This decomposition emerges purely from task-level supervision without any explicit prototype assignment, demonstrating that structured action representations can be self-organized through the proposed gating mechanism.

D MCF Axis Alignment Analysis

To examine how the learned frame organizes multi-stage manipulation, Figure 7 shows the per-timestep distribution of the dominant axis—the MCF axis carrying the largest action magnitude—aggregated across episodes of *pick up the book and place it in the black compartment of the caddy*.

The resulting axis transitions reveal how different task phases are expressed within the learned local frame. The X/Y/Z labels refer to the displayed model's coordinate convention.

The distribution reveals a clear phase structure aligned with the task's semantic stages. During the early phase ($t \approx 0$–25), actions are predominantly decomposed along the MCF-Y (green) axis, corresponding to downward reaching toward the book. In the grasping and posture adjustment phase ($t \approx 25$–35), the MCF-Z (blue) axis dominates as the robotic arm rotates about its axis after securing the book. During the lateral translation phase ($t \approx 35$–45), the MCF-X (red) axis governs the motion of moving the book toward the caddy. In the final phase ($t \approx 45$–50), the MCF-Y (green) axis again dominates, corresponding to the book-placement movement. An overlap between Phase 2 and Phase 3 indicates that the robotic arm performs simultaneous translation and rotation, which is also consistent with empirical observations.

Crucially, this axis-switching behavior emerges without any explicit phase annotation or axis supervision. Rather than indicating that the frame rotates to follow the instantaneous motion direction at every step, this pattern suggests that the MCF evolves smoothly over time, while stage changes are primarily expressed through shifts in which axis becomes dominant. This interpretation is consistent with the smoothness regularization described in the main text, which encourages temporal consistency in the frame and explains multi-stage changes through axis reassignment rather than excessive frame rotation. Overall, the smooth yet structured transitions between dominant axes suggest that the MCF provides a principled decomposition of multi-stage manipulation into a sequence of locally near-uniaxial motions, supporting more compact and reusable action representations.

E Additional visualization

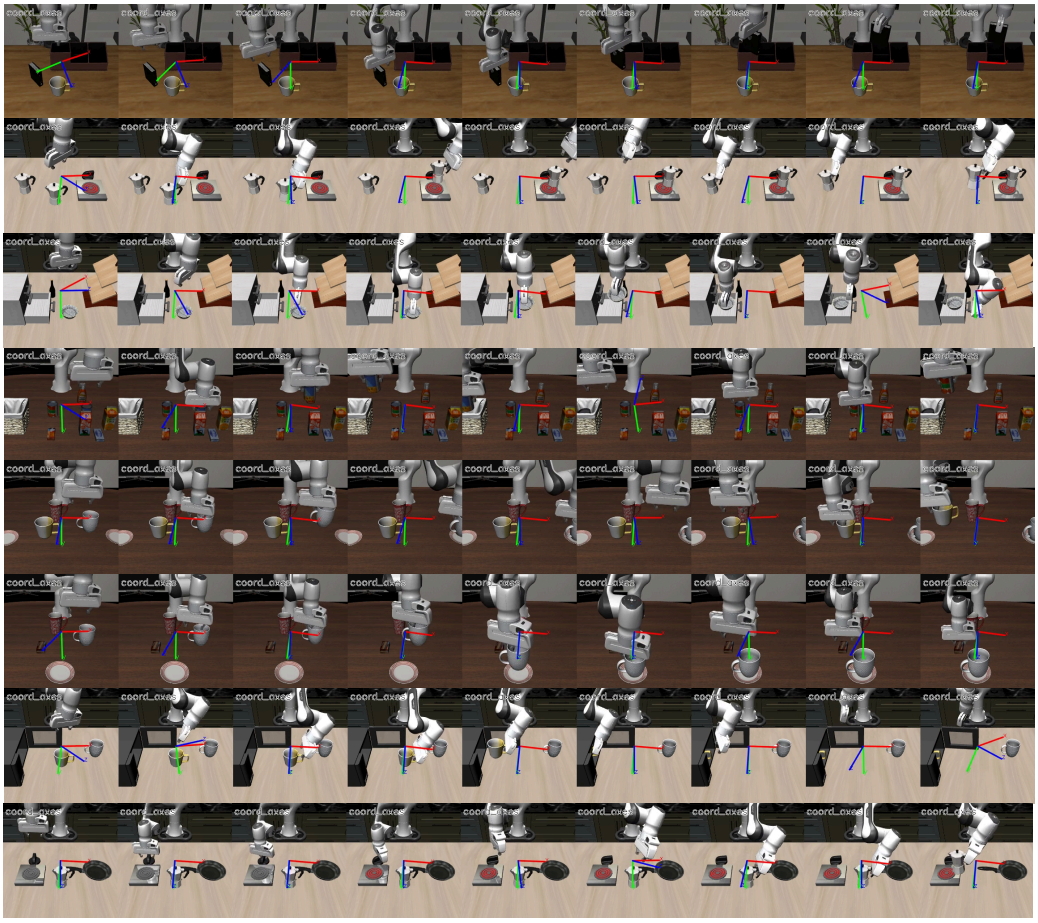

Figure 8: Additional visualization results of MCF-Proto

Table 9: Prototype-count sensitivity on LIBERO with $d = 3$, and head cost relative to the matched MLP. Latency is per action chunk on identical profiling hardware.

Prototype count K	4	8	16	32
LIBERO success (%)	95.6	97.2	97.7	97.6

Head	Parameters	Relative latency
Matched MLP	approximately 42.0M	$1.000\times$
MCF-Proto	approximately 42.2M	$1.021\times$

F Prototype count and computational cost

Table 9 shows that performance plateaus around $K = 16$; increasing to 32 brings no further improvement in this evaluation. We use $d = 3$ to represent three-dimensional translation and axis-angle increments, rather than treating this as a strict dimensionality reduction of each individual increment. The prototype dictionaries contain $2 \times 16 \times 3 \times 3 = 288$ parameters. The measured head overhead is approximately 0.2M parameters and 2.1% per-chunk inference latency relative to the matched MLP head.

G Backbone transfer

With the same MCF-Proto head design and a PaliGemma-3B backbone pretrained for $\pi_{0.5}$ [34], LIBERO-plus success rates are 72.7, 69.5, 80.6, 98.6, 94.2, 85.3, and 84.1 percent for Camera, Robot, Language, Light, Background, Noise, and Layout, respectively (mean 83.6%, versus 81.9% with Qwen3-VL-4B). This evaluates transfer of the head across backbones; the difference also includes the effect of robot pretraining and does not isolate architecture alone.

H Additional real-world protocol and cross-platform results

The OpenArm table reports success rates for completing at least the first k tube placements, together with the average number of completed tubes out of four. The bimanual-transfer and bowl-stacking columns report full-task success.

We also evaluate the three tasks on AgileX Piper. On the four-tube task, success rates at stages 1–4 are 96.0, 84.0, 74.0, and 66.0 percent for MCF-Proto, compared with 92.0, 70.0, 54.0, and 44.0 percent for $\pi_{0.5}$. The corresponding completed counts are 3.20 and 2.60; full-task success on block transfer is 88.0 percent versus 76.0 percent, and bowl stacking is 90.0 percent versus 84.0 percent. These results show improved task completion across the two platforms.