arXiv:2607.11884v2 [cs.RO] 28 Oct 2026

Mixture of Frames Policy: Multi-Frame Action Denoising for Bimanual Mobile Manipulation

Dian Wang* **Jisang Park*** **Xiaomeng Xu** **Han Zhang** **Shuran Song†** **Jeannette Bohg†**
Stanford University
https://mofpo.github.io

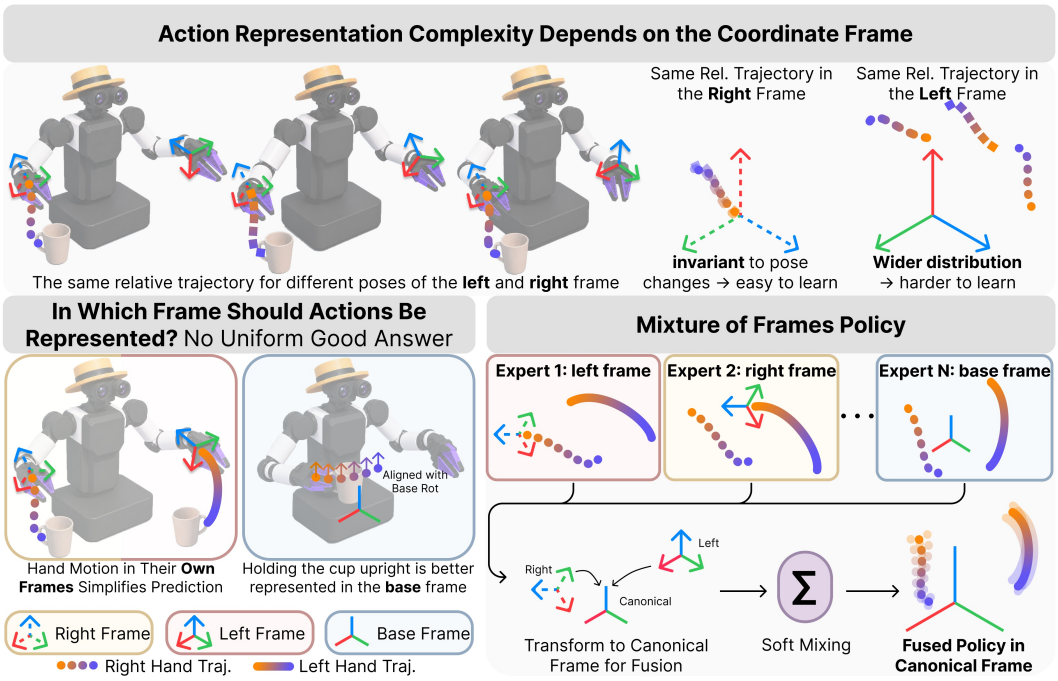


Figure 1: **Mixture of Frames Policy (MoF).** The complexity of a manipulation action distribution depends strongly on the coordinate frame in which it is represented. The same relative motion can be compact and invariant in one frame (e.g. an end-effector frame), yet form a broader and harder-to-learn distribution in another. Conversely, some motions, such as keeping an object upright, are better represented in a base-aligned frame. Since no single frame is universally optimal, MoF trains frame-specialized experts and fuses their predictions in a shared canonical frame.

Abstract: Robotic manipulation is inherently multi-frame: local actions may be simple in an end-effector frame, while transport, upright-object handling, and whole-body coordination are better represented in a base-aligned frame. However, modern diffusion-based visuomotor policies typically commit to a single predefined action frame, forcing one denoiser to model action distributions that are often unnecessarily complex in that frame. We propose Mixture of Frames Policy (MoF), a diffusion policy that performs synchronized action denoising across multiple coordinate frames. MoF maintains a single canonical diffusion state, re-expresses it in several task-relevant frames, applies frame-specialized denoisers, and fuses their noise predictions back in the canonical frame. To make this possible for intermediate noisy diffusion states, we adopt a column-based 6D rotation representation within an SE(3) action parameterization that supports exact, differentiable frame transformations without requiring noisy rotations to lie on the SO(3) manifold.

* Indicates equal contribution, † indicates equal advising.

10th Conference on Robot Learning (CoRL 2026), Austin TX, USA.

Across nine simulated bimanual manipulation tasks, we show that the best action frame is task-dependent and that MoF improves over oracle frame selection and standard Mixture-of-Experts (MoE) baselines. We further evaluate MoF on two real-world bimanual mobile manipulation tasks, demonstrating that it outperforms all constituent single-frame baselines.

Keywords: Bimanual Mobile Manipulation, Visuomotor Policy Learning

1 Introduction

Robotic manipulation systems are inherently multi-frame. A bimanual mobile robot naturally operates with a base frame, left and right end-effector frames, camera frames, and other task-relevant coordinate frames defined by its kinematics and embodiment. These frames are not only alternative parameterizations of the same state, but can also make different parts of the same task easier to model. For example, the action of approaching and grasping a cup handle is more naturally expressed in the gripper frame: if the cup and gripper appear in the same relative configuration, the desired gripper frame action is invariant, even if the gripper-object pair is in different absolute poses relative to the base frame (Fig. 1 top). In contrast, when the robot transports a filled cup, maintaining the cup upright and moving it coherently with the mobile base is more naturally expressed in the base frame.

This frame choice matters because different tasks can be easier to learn when the action is represented in a specific frame. However, modern visuomotor policies such as Diffusion Policies [1] typically choose a single predefined coordinate frame to parameterize the action. Even when transformations among frames are provided as proprioceptive observations, the denoiser must still model the entire action distribution in one chosen frame. Moreover, the most useful frame might vary while a manipulation task is progressing. A single-frame policy therefore inherits whatever complexity the chosen frame imposes on the action distribution, regardless of whether that complexity is intrinsic to the task or merely an artifact of the parameterization.

Reasoning over multiple coordinate frames offers a natural solution to this limitation, allowing the policy to use each frame when the corresponding action distribution is easiest to model. Prior work has explored multi-frame reasoning in robot learning by mining observation frames for point-cloud policies [2], and composing low-dimensional frame-conditioned skills from demonstrations in non-visual, state-based skill learning [3]. However, multi-frame reasoning has remained largely under-explored in the context of modern diffusion-based visuomotor policies.

This suggests that frames should not only be provided as observations, but should also define alternative action spaces in which denoising can be performed. In this paper, *we propose Mixture of Frames policy (MoF), a diffusion policy that denoises actions in multiple coordinate frames in parallel*. Instead of using a single denoiser in one predefined coordinate frame, MoF instantiates one expert denoiser per reference frame and fuses their predictions at each denoising step through either a mixture-of-experts pipeline or simple ensembling. This allows the policy to reason over multiple coordinate frames simultaneously, rather than committing to a fixed frame selected by the designer. To our knowledge, MoF is the first diffusion-based visuomotor policy that performs denoising directly across multiple action frames. Our contributions can be summarized as follows:

- We present a systematic analysis showing that the choice of coordinate frame has a substantial and task-dependent effect on diffusion policy performance in bimanual mobile manipulation, motivating policies that reason natively across multiple frames.
- We propose Mixture of Frames (MoF) Policy, a diffusion policy architecture that performs synchronized action denoising in multiple coordinate frames in parallel, with per-frame expert denoisers and a learned router or uniform weighting that combines their predictions.
- We adopt a column-vector $\mathrm{SE}(3)$ action representation that admits exact, differentiable transformation of both noisy and noise-free actions across reference frames, facilitating consistent multi-frame denoising.

2 Related Works

Learning Bimanual Mobile Manipulation Bimanual mobile manipulation requires learning coordinated whole-body motion. Prior works have addressed bimanual coordination by decomposing roles [4] or modeling inter-arm interactions [5, 6, 7], while mobile manipulation approaches often decouple whole-body coordination from policy learning [8, 9, 10]. Most recent bimanual mobile manipulation works address both via co-training with fixed-base data [11, 12] or introducing specialized architectures [13, 14]. However, these approaches typically rely on a single frame representation, which may be insufficient when coordination requirements shift throughout task progression.

Multi-Frame Reasoning for Manipulation Manipulation policies benefit from multi-frame reasoning as different decisions are most naturally expressed in distinct coordinate systems [15]. Earlier task-parameterized methods address this by composing multiple local frames via generalized weighting [16] or time-varying relevance estimation [3]. Subsequent learning-based works introduce adaptive frame selection in point-cloud policies [2] and frame-based interfaces in hierarchical control [17, 18]. In parallel, recent bimanual manipulation works adopt relation-centric representations to encode hand-object and arm-arm geometries [19, 20, 7]. These studies motivate the adaptive fusion of frame-specialized experts, as optimal frames vary by subproblem.

Mixture-of-Experts for Manipulation Mixture-of-Experts (MoE) [21, 22] enables modular experts to capture diverse local behaviors more effectively than monolithic policies in robot learning [23, 24]. Existing MoE-based manipulation policies specialize experts along behavioral dimensions such as task identity [25, 26], skill abstraction [27, 28], and execution phases [29], or architectural components such as denoising stages [30] and sensing modalities [31]. We instead formulate MoE over coordinate frames, addressing the underexplored challenge of multi-frame reasoning through the adaptive fusion of frame-specific action representations.

3 Method

We present **Mixture of Frames** (MoF) Policy, a diffusion policy that denoises a single action trajectory through multiple coordinate-frame parameterizations. In this section, we first present the problem statement, then introduce a synchronized multi-frame denoising procedure, followed by a transformation-compatible action representation that makes this procedure exact for noisy actions.

3.1 Problem Statement

We consider visuomotor imitation learning for bimanual mobile manipulation with diffusion policies [1]. At each control step, the policy receives an observation $o = (\mathcal{I}, q)$, where $\mathcal{I}$ denotes RGB images and q denotes proprioception. The policy predicts an action chunk $a \in \mathbb{R}^{\tau \times d}$ over horizon τ. In our setting, each action timestep contains the target poses of the left and right end-effectors and the gripper commands. These end-effector targets are tracked by a whole-body controller for execution. We denote $\mathcal{F} = \{F_m\}$ as a family of task-relevant coordinate frames, including but not limited to the left and right end-effector frames F_l and F_r, and the base frame F_b. Given proprioception, the rigid transforms from any frame m to another frame m', ${}^{m'}T_m \in \mathrm{SE}(3)$, are known.

In standard diffusion policies, the designer fixes a reference action frame $F^\star \in \mathcal{F}$, and both training targets and denoising predictions are represented in this frame. For example, prior works have used the world frame [1, 32, 33], the end-effector frame [34, 35], or the left-hand frame [14]. However, it is unclear which single fixed $F^\star$ is universally best for bimanual mobile manipulation. This motivates a policy that maintains multiple frame-specialized denoisers and combines their predictions adaptively.

3.2 Consistent Multi-Frame Denoising

A straightforward way to use multiple frame-specialized denoisers (i.e., experts) would be to run an independent diffusion process in each frame and fuse the final denoised actions. However,

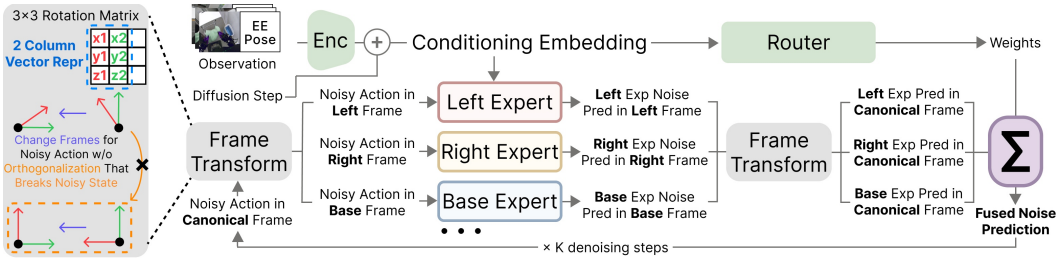


Figure 2: **MoF Architecture.** At each diffusion step, MoF maintains a noisy action in the canonical frame. This noisy action is re-expressed in each expert's frame, and all experts predict noise in their native frame. The predicted noise vectors are then transformed back to the canonical frame and fused using either router-predicted weights or fixed uniform weights. The fused noise prediction is used by a standard denoising step, keeping all frame experts synchronized along the same diffusion trajectory. Left: to enable frame change for noisy actions, we employ a column-vector action representation where frame transform can be achieved without orthogonalization, preserving the noisy action state.

this is problematic with multimodal action distributions, as different denoisers may converge to different modes, making their final action predictions incompatible. Instead, MoF keeps all experts synchronized by maintaining a single diffusion trajectory in a canonical frame and fusing predictions at every denoising step, as shown in Fig. 2.

Specifically, let F_c denote the canonical frame in which the scheduler state x_c^k is stored. For each frame $F_m \in \mathcal{F}$, we transform the canonical-frame noisy action sample x_c^k into the expert frame via ${}^m\mathcal{T}_c(\cdot)$, feed it to a frame-specialized denoiser ϵ_m^θ, transform the predicted noise back to the canonical frame using ${}^c\mathbf{T}_m(\cdot)$, and fuse the aligned predictions:

$$\hat{\epsilon}_c = \sum_{F_m \in \mathcal{F}} w_m \, {}^c\mathbf{T}_m\big(\epsilon_m^\theta\left({}^m\mathcal{T}_c(x_c^k), o; k\right)\big) . \tag{1}$$

Here, both $\mathcal{T}$ and $\mathbf{T}$ are induced by the rigid transform between F_c and F_m, and their concrete definitions are given in Sec. 3.3. w_m is the weight for the denoiser in frame m, which can be either set by the designer or learned through a router, with $\sum_m w_m = 1$. This synchronized update has two benefits. First, all experts contribute to the same underlying diffusion trajectory, avoiding the mode-mismatch problem. Second, because fusion occurs in the canonical noise space, the diffusion scheduler remains unchanged: MoF can be implemented on top of a standard DDIM sampler by replacing the single noise prediction with the fused multi-frame prediction $\hat{\epsilon}_c$.

Training follows the same principle. We sample a canonical-frame action $x_c^0 = a$, noise $\epsilon_c \sim \mathcal{N}(0, I)$, and diffusion step k, construct x_c^k the same way as standard Diffusion Policy, and train the fused canonical prediction to match the canonical noise:

$$\mathcal{L}_{\text{mix}}(\theta) = \mathbb{E}_{k,\epsilon_c,o}\left[\left\|\epsilon_c - \sum_{F_m \in \mathcal{F}} w_m \, {}^c\mathbf{T}_m\big(\epsilon_m^\theta\left({}^m\mathcal{T}_c(x_c^k), o; k\right)\big)\right\|^2\right] . \tag{2}$$

Thus, the training target remains the standard diffusion noise in the canonical frame, while each expert learns to denoise the same underlying noisy action expressed in its own frame.

3.3 Transformation-Compatible Action Representation

The synchronized denoising procedure in Sec. 3.2 requires frame transformations to be applied directly to intermediate diffusion states, whose rotation channels generally do not form a valid element of $\mathrm{SO}(3)$. This poses a challenge for the row-vector 6D rotation representation used in standard Diffusion Policy implementations, which stores the first two rows of the rotation matrix. Changing frames under this representation requires reconstructing a valid rotation matrix via orthogonalization before applying the transform. This projection is nonlinear and lossy: it replaces the noisy rotation channels with a projected rotation, thereby changing the diffusion state itself.

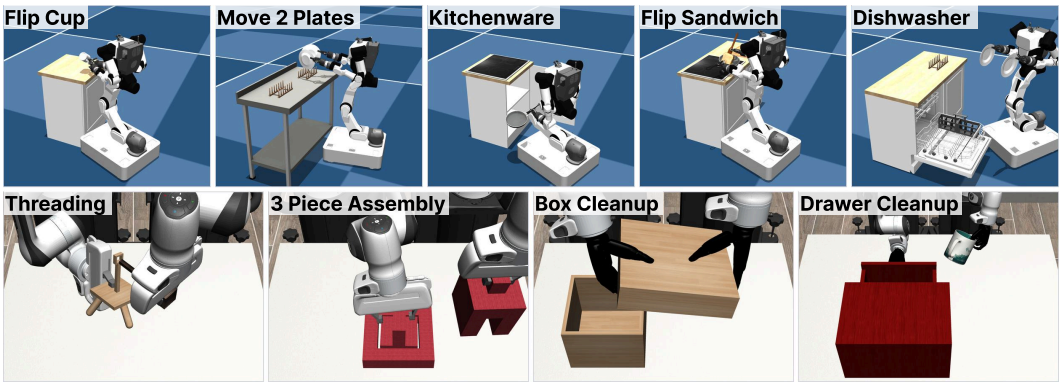


Figure 3: **Simulation Tasks.** Nine tasks from BiGym [36] (top) and DexMimicGen [37] (bottom).

Table 1: Single-frame DP baselines (%, mean ± std. err. over 3 seeds). **Oracle** selects the best frame per task (bold). **Worst** selects the worst per task, with parenthetical values denoting Worst − Oracle.

Frame	Avg	Flip Cup	Move 2 Plates	Kitchenware	Flip Sandwich	Dishwasher	Threading	3 Piece Assembly	Box Cleanup	Drawer Cleanup
Left	55.6	47.3 ± 1.0	44.5 ± 3.1	29.2 ± 3.5	$\mathbf{36.1 \pm 3.3}$	78.8 ± 2.5	43.9 ± 4.4	51.5 ± 6.8	$\mathbf{92.4 \pm 1.2}$	76.8 ± 1.7
Right	55.0	45.1 ± 1.9	$\mathbf{47.2 \pm 1.7}$	22.5 ± 0.7	34.3 ± 1.6	83.1 ± 0.5	34.7 ± 3.5	69.1 ± 2.9	79.9 ± 7.2	79.6 ± 1.5
Base	61.0	$\mathbf{53.9 \pm 2.9}$	29.7 ± 0.1	$\mathbf{31.2 \pm 2.6}$	35.5 ± 0.3	$\mathbf{84.1 \pm 1.6}$	62.3 ± 3.0	$\mathbf{76.3 \pm 0.5}$	86.1 ± 6.3	$\mathbf{90.1 \pm 1.0}$
Rel Traj	55.8	42.0 ± 2.1	32.1 ± 2.6	26.1 ± 1.9	25.5 ± 3.2	76.4 ± 0.6	$\mathbf{63.1 \pm 0.7}$	73.7 ± 2.3	80.0 ± 2.3	83.5 ± 1.0
Oracle Frame	63.8	53.9	47.2	31.2	36.1	84.1	63.1	76.3	92.4	90.1
Worst Frame	48.8 (-15.0)	42.0 (-11.9)	29.7 (-17.5)	22.5 (-8.7)	25.5 (-10.6)	76.4 (-7.7)	34.7 (-28.4)	51.5 (-24.8)	79.9 (-12.5)	76.8 (-13.3)

We instead use a column-vector rotation representation. For one robot arm at one action timestep, the pose/action channels expressed in the canonical frame F_c are $x_c = [p_c, c_c^1, c_c^2, g]$, where $p_c \in \mathbb{R}^3$ is the end-effector position, $c_c^1, c_c^2 \in \mathbb{R}^3$ are the first two columns of the end-effector orientation, and g is the gripper command. For a clean end-effector orientation, let ${}^cR_{ee} = [c_c^1, c_c^2, c_c^3]$ denote the rotation from the end-effector frame to the canonical frame. Re-expressing the same orientation in an expert frame F_m uses the known frame-change rotation ${}^mR_c$, giving ${}^mR_{ee} = {}^mR_c{}^cR_{ee} = [{}^mR_c c_c^1, {}^mR_c c_c^2, {}^mR_c c_c^3]$. Thus, each stored column transforms as $c_m^i = {}^mR_c c_c^i$. This operation is well-defined even when c_c^1 and c_c^2 are arbitrary noisy vectors rather than columns of a valid rotation matrix. Therefore, changing action frames only requires left-multiplying ordinary 3D vectors by the frame-change rotation ${}^mR_c$, without projecting onto $\mathrm{SO}(3)$, as illustrated in Fig. 2 left.

Let ${}^mT_c$ denote the rigid transform from the canonical frame F_c to an expert frame F_m. This transform induces the action-state operator ${}^m\mathcal{T}_c$ and the noise-vector operator ${}^m\mathbf{T}_c$ used in Eq. (1). Their explicit definitions differ only in whether the translation offset is applied to position channels and are provided in Appendix B.1. With this representation, MoF can transform noisy action states and predicted noise vectors across frames without projecting noisy rotations back onto $\mathrm{SO}(3)$.

4 Simulation Experiments

Our simulation experiments aim to answer the following questions:

- **Q1.** How much does the choice of action frame affect diffusion-policy performance?
- **Q2.** Can MoF match the best single-frame policy and outperform existing baselines?
- **Q3.** Which components of MoF are responsible for the performance improvement?

We evaluate MoF on nine tasks across BiGym [36] (*FlipCup*, *Move2Plates*, *Kitchenware*, *FlipSandwich*, *Dishwasher*) and DexMimicGen [37] (*Threading*, *3PieceAssembly*, *BoxCleanup*, *DrawerCleanup*). All policies are trained for 500 epochs with 100 demonstrations and three random seeds. For each seed, we evaluate five checkpoints from epochs 460–500 at 10-epoch intervals using 50 episodes each. We report the mean success rate (%) and standard error. See Appendix C for details.

Q1. Evaluation of Frame Choice We first ask whether action-frame choice is an important design decision for diffusion policies, and whether a single hand-designed frame is sufficient across tasks.

Table 2: MoF vs. baselines: last-5 success rate (%, mean ± std. err. over 3 seeds). Bold indicates best, underline indicates second.

Method	Avg	Flip Cup	Move 2 Plates	Kitchenware	Flip Sandwich	Dishwasher	Threading	3 Piece Assembly	Box Cleanup	Drawer Cleanup
MoF MoE (Ours)	**66.8**	56.5 ± 1.4	**51.6 ± 4.4**	31.6 ± 2.4	40.0 ± 3.1	89.2 ± 3.9	70.8 ± 7.5	77.3 ± 2.0	87.1 ± 3.2	**96.9 ± 0.7**
MoF Ens. (Ours)	65.9	**59.2 ± 0.4**	51.5 ± 1.5	**33.7 ± 0.5**	**43.2 ± 2.0**	**89.7 ± 1.4**	68.7 ± 3.3	76.0 ± 1.3	77.5 ± 5.2	93.3 ± 2.9
Oracle Frame	63.8	53.9 ± 2.9	47.2 ± 1.7	31.2 ± 2.6	36.1 ± 3.3	84.1 ± 1.6	63.1 ± 0.7	76.3 ± 0.5	**92.4 ± 1.2**	90.1 ± 1.0
MF-DP	60.3	54.0 ± 1.6	34.5 ± 5.1	22.7 ± 3.8	38.8 ± 2.4	89.6 ± 2.2	69.7 ± 0.7	70.0 ± 2.4	68.4 ± 2.1	95.1 ± 0.7
Single-Frame Ens.	55.9	46.4 ± 3.6	31.2 ± 2.7	25.9 ± 3.1	31.2 ± 2.8	74.5 ± 3.6	69.7 ± 2.2	75.7 ± 1.7	66.5 ± 3.7	81.7 ± 2.7
Sym-in-DP [34]	57.6	46.3 ± 1.9	31.7 ± 1.1	26.8 ± 1.6	24.3 ± 3.7	81.7 ± 2.1	63.1 ± 4.2	**78.4 ± 2.3**	79.9 ± 4.4	86.5 ± 1.5
MoE-DP [29]	55.1	32.9 ± 3.8	27.9 ± 1.7	18.7 ± 0.8	29.9 ± 2.0	72.7 ± 0.9	**72.5 ± 5.2**	73.5 ± 2.6	74.7 ± 6.0	93.5 ± 1.9
DP [1]	50.3	17.2 ± 2.8	38.5 ± 0.4	22.0 ± 1.6	32.5 ± 3.1	61.5 ± 2.1	47.9 ± 1.9	66.7 ± 1.5	73.3 ± 8.9	92.9 ± 0.9

To answer this, we train single-frame diffusion policies whose action representation is fixed to one of the four candidate expert parameterizations used by MoF (See Appendix A.1 for details): left end-effector, right end-effector, base-relative, and per-arm relative-trajectory. We also report two aggregate references. Oracle Frame selects the best single frame for each task after evaluation and therefore represents the best possible task-level frame selection among our candidates. Worst Frame selects the worst single frame for each task, quantifying the cost of an unfavorable frame choice.

Table 1 shows that action frame selection has a large and task-dependent impact on diffusion policy performance. Among the individual frames, the base frame obtains the highest average success rate, but it is not uniformly best across tasks. It performs well on five tasks, while the left frame is best on two tasks, the right frame and the relative trajectory frame are each best on one task. The gap between the oracle and worst frame further quantifies this sensitivity. On average, the worst frame is 15%p below the oracle frame. These results confirm that no single hand-designed frame consistently captures the easiest action distribution across all bimanual manipulation tasks.

Q2. Comparison of MoF and Baselines We next ask whether our method improves over single-frame baselines that do not reason over action frames. We evaluate two variants of MoF. **MoF-MoE**, shown in Fig. 2, uses a router conditioned on the diffuser-conditioning embedding to predict the expert weights w_m. **MoF-Ensemble** removes the learned router and uses uniform weights, $w_m = 1/|\mathcal{F}|$, effectively averaging the aligned predictions from all frame experts. We compare MoF-MoE and MoF-Ensemble with six baselines. **Oracle Frame** is the strongest single-frame reference from Table 1. Comparing against it tests whether MoF automates frame selection or potentially does more than that. **DP** [1] is the standard Diffusion Policy trained in a single world-frame action representation, measuring improvement over the conventional design. **MoE-DP** [29] uses a mixture-of-experts module in the conditioning pathway of DP, but still denoises actions in a single action representation. Comparing against it tests whether the gains come from frame-level denoising rather than a generic MoE architecture. We compare against **Single-Frame Ensemble** and **MF-DP** to assess the need for frame-specialized denoisers: the former averages four same-frame diffusers, while the latter uses a capacity-matched single denoiser with multi-frame inputs. In the end, we compare against **Sym-in-DP** [34], a geometry-aware baseline leveraging rotational equivariance in the vision encoder.

Table 2 shows that both MoF variants outperform the single-frame baselines on average. MoF-MoE achieves an average success rate of 66.8%, improving over MoE-DP by 11.7 percentage points and over standard DP by 16.5 percentage points. MoF-Ensemble achieves a similar average success rate of 65.9%, while the same-frame Ensemble baseline performs substantially worse. This indicates that multi-frame denoising, rather than simply using multiple diffusers, is the main source of improvement. MoF-MoE also exceeds capacity-matched MF-DP by 6.5 percentage points, showing that multi-frame inputs alone are insufficient and that frame-specialized denoisers are needed to

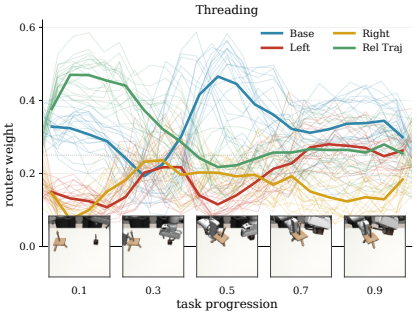


Figure 4: **Router Weights on Threading.** For each MoF-MoE expert, faint lines show router weights across task progression for individual episodes, and bold lines show the 20-episode mean.

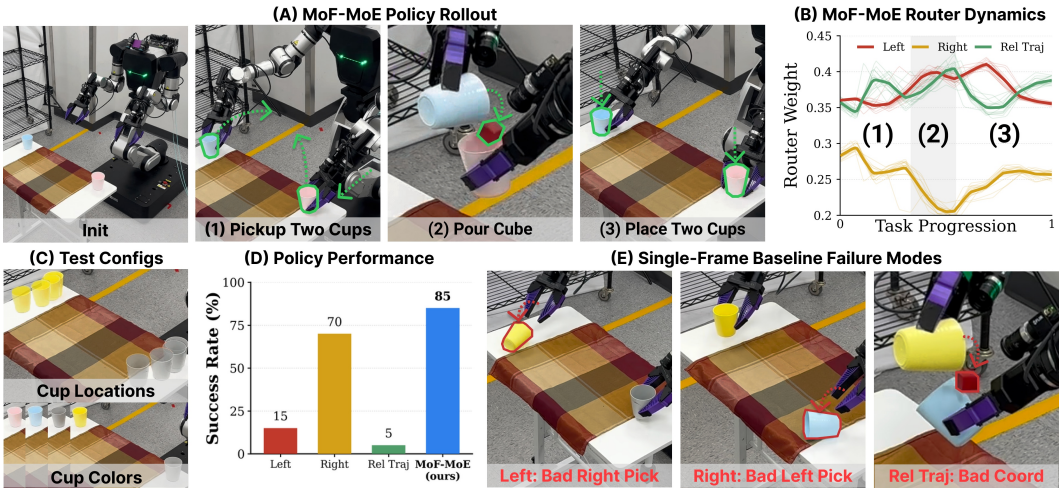


Figure 5: **Pouring Task.** (a) Our MoF policy rollout, (b) average router weights across successful rollouts relative to task progression, (c) evaluation configurations with different cup locations and color combinations, (d) overall performance, and (e) failure cases of each single-frame baseline.

fully exploit multiple action representations. MoF-MoE also outperforms geometry-aware Sym-in-DP by 9.2 percentage points, supporting the benefit of frame-specialized denoising.

MoF also matches or outperforms Oracle Frame on most tasks. The largest improvement is on Threading (+7.7 points). To probe where this gain comes from, we run the trained MoF-MoE policy over 20 training demonstrations and record the router's expert weights at each denoising step. Fig. 4 plots these weights, averaged over the last 4 denoising steps, as a function of task progression. The rel-traj expert is the dominant frame during the first part of the episode, where each gripper reaches the two objects, a motion that is naturally compact in an arm-centric frame. The base-relative expert takes over for the second part, during the coordinated bimanual insertion that requires consistent reasoning across both arms. This phase-dependent frame usage is not available to any single-frame baseline, and is consistent with MoF-MoE exceeding even Oracle Frame on this task. (See Appendix D)

Q3. Ablation Study Finally, we ablate MoF design choices on four representative tasks: *FlipCup*, *Move2Plates*, *Threading*, and *Box-Cleanup*. Table 3 shows the average success rates. Changing the canonical frame has only a modest effect, with the largest drop being 2.5 points, suggesting that MoF is robust to the arbitrary choice of canonical frame. In contrast, removing the best-performing single frame from the expert set reduces performance by 6.2 points, indicating that the expert frame set is not redundant. Applying orthogonalization before frame transformation reduces MoF-MoE by 7.0 points and collapses MoF-Ensemble to 0.0%, validating the need for the transformation-compatible action representation in Sec. 3.3. See Appendix E for full results and Appendix F for additional experiments on expert frame selection and scalability.

Table 3: Ablation avg. over four tasks. Parentheses denote differences from ours.

Method	Avg
MoF MoE (Ours)	66.5
Rel Traj as Canonical	64.0 (−2.5)
Left as Canonical	65.8 (−0.7)
Right as Canonical	66.4 (−0.1)
w/o Best Frame	60.3 (−6.2)
w/ Ortho Proj	59.5 (−7.0)
MoF Ensemble (Ours)	64.2
w/ Ortho Proj	0.0 (−64.2)

5 Real World Experiments

Setup Our real-world experiments study how multi-frame fusion improves robustness over single-frame policies in bimanual mobile manipulation. Specifically, our tasks require dynamic switching between multi-frame reasoning modes: global navigation, bimanual coordination, and decoupled manipulation. We integrate MoF-MoE into the HoMMI [14] real-world framework. As the HoMMI data collection interface lacks base frame tracking, our MoF-MoE policy uses three experts—the left, right, and relative-trajectory experts—and we compare it against the corresponding single-frame

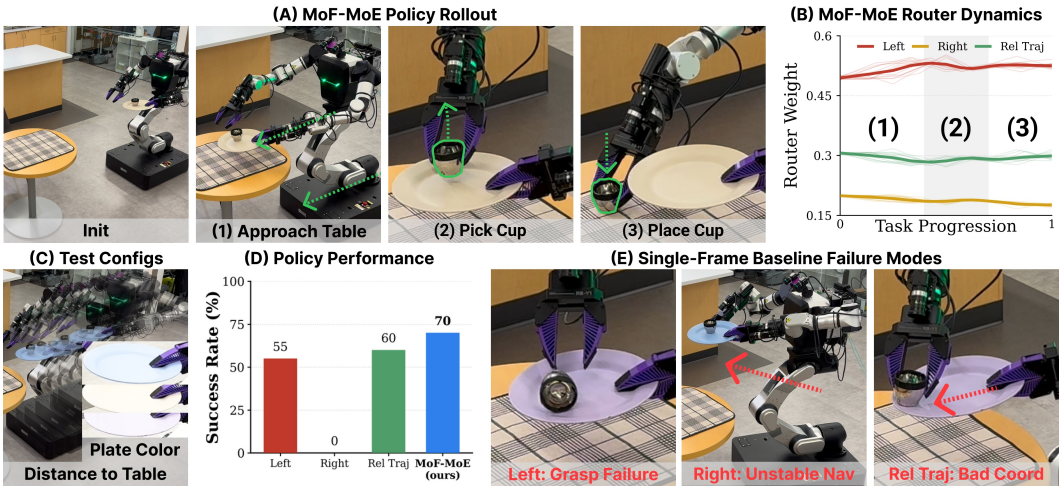


Figure 6: **Serving Task.** (a) Our MoF policy rollout, (b) average router weights across successful rollouts relative to task progression, (c) evaluation configurations with varying table distances and seen/unseen plate colors, (d) overall performance, and (e) failure cases of each single-frame baseline.

baselines. For each task, we evaluate each method over 20 rollouts with different initial configurations and report task success rates. See Appendix G for details and deeper analysis of results.

Pouring Task As shown in Fig. 5, the robot must approach and pick up two cups, pour a cube from the right to the left cup, and place both upright on the table. This requires transitioning between local end-effector-centric reasoning for precise pick-and-place and tight bimanual coordination for pouring. Each policy is trained on 257 demonstrations and evaluated in 20 rollouts across 5 color combinations (3 seen, 2 unseen) and 4 cup placements. Our `MoF-MoE` achieves 85% success, with only 3 failures in picking up (1 left, 2 right). Router analysis reveals `Left` and `Rel Traj` alternating during pick-and-place, with `Left` dominating during pouring as the static left cup provides a more stable reference than the moving right arm. Single-frame baselines fail at distinct phases: (1) `Left` (15%) misses right cup pickup and fails during pouring. (2) `Right` (70%) fails left cup pickup and placement. (3) `Rel Traj` (5%) struggles with left cup pickup or pouring due to unstable grasp poses.

Serving Task As shown in Fig. 6, the robot must navigate while balancing a plate, grasp a cup, and place it upright on the table. This requires stable single-arm transport of the plate during approach, transitioning to bimanual coordination for grasping and local end-effector-centric reasoning for placement. Each policy is trained on 177 demonstrations and evaluated in 20 rollouts across 3 plate colors (1 seen, 2 unseen) and varying table distances. Our `MoF-MoE` achieves 70% success, with four grasping failures and two placement failures. Router weights show `Left` consistently dominating while `Right` remains the lowest, indicating that reasoning relative to the stably held plate is optimal throughout the task. Conversely, single-frame baselines exhibit distinct failure modes: (1) `Left` (55%) causes the cup to shift on the plate over extended navigation distances, resulting in grasp failures. (2) `Right` (0%) exhibits highly unstable navigation, failing even to approach the grasp phase. (3) `Rel Traj` (60%) navigates stably but lacks bimanual coordination during the grasping phase.

6 Conclusion

We present Mixture of Frames Policy (MoF), a diffusion-policy framework for denoising actions across multiple coordinate-frame parameterizations. Our experiments show that action-frame choice has a substantial and task-dependent effect on diffusion-policy performance, and that fusing frame-specialized denoisers improves over baselines including oracle task-level frame selection.

7 Limitations

This work has several limitations. First, while theoretically adaptable to larger vision-language-action (VLA) models, our current experiments focus on smaller diffusion policies. Scaling to VLA backbones remains an important future direction. Second, the router is supervised solely via denoising loss, which might not strictly correspond to policy performance. Future work could explore behavior-aware objectives that directly optimize execution quality. Third, MoF relies on a designer-specified set of candidate frames and accurate frame transformations from robot proprioception. Future work could investigate learning or automatically discovering task-relevant frames and study how frame uncertainty affects multi-frame denoising in real-world deployment. Finally, MoF currently relies on frame-dependent Cartesian action representations and does not directly extend to purely joint-space or torque-based policies, whose actions are invariant to coordinate-frame changes.

Acknowledgments

The authors would like to thank Yifan Hou for the help during RB-Y1 controller implementation, and Mengda Xu for the helpful discussion during project brainstorming. This work utilized Marlowe, Stanford's GPU-based computational instrument [38]. This work was supported in part by the NSF Award #2143601, #2037101, and #2132519, Apple, and Toyota Research Institute. Dian Wang is supported by the Stanford HAI PostDoc fellowship. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the sponsors.

References

[1] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

[2] M. Liu, X. Li, Z. Ling, Y. Li, and H. Su. Frame mining: a free lunch for learning robotic manipulation from 3d point clouds. In *Conference on Robot Learning*, pages 527–538. PMLR, 2023.

[3] M. R. Montero, G. Franzese, J. Kober, and C. Della Santina. Learning multi-reference frame skills from demonstration with task-parameterized gaussian processes. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2832–2839. IEEE, 2024.

[4] J. Grannen, Y. Wu, B. Vu, and D. Sadigh. Stabilize to act: Learning to coordinate for bimanual manipulation. In *Conference on robot learning*, pages 563–576. PMLR, 2023.

[5] A. C.-W. Lee, I. Chuang, L.-Y. Chen, and I. Soltani. Interact: Inter-dependency aware action chunking with hierarchical attention transformers for bimanual manipulation. In *Conference on Robot Learning*, pages 1730–1743. PMLR, 2025.

[6] J.-J. Jiang, X.-M. Wu, Y.-X. He, L.-A. Zeng, Y.-L. Wei, D. Zhang, and W.-S. Zheng. Rethinking bimanual robotic manipulation: Learning with decoupled interaction framework. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12427–12437, 2025.

[7] Z. Chen, N. T. Chan, Y. Hou, C. Tie, Z. Liu, H. Chen, J. Chen, J. Shi, Y. Gao, J. Huo, and L. Shao. Rotri-diff: A spatial robot–object triadic interaction-guided diffusion model for bimanual manipulation. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, pages 21530–21537. IEEE, 2026.

[8] J. Yang, I. Huang, B. Vu, M. Bajracharya, R. Antonova, and J. Bohg. Mobi-π: Mobilizing your robot learning policy. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 3516–3536. PMLR, 27–30 Sep 2025.

[9] H. Ha, Y. Gao, Z. Fu, J. Tan, and S. Song. Umi-on-legs: Making manipulation policies mobile with manipulation-centric whole-body controllers. In *Conference on Robot Learning*, pages 5254–5270. PMLR, 2025.

[10] P. Sundaresan, R. Malhotra, P. Miao, J. Yang, J. Wu, H. Hu, R. Antonova, F. Engelmann, D. Sadigh, and J. Bohg. Homer: Learning in-the-wild mobile manipulation via hybrid imitation and whole-body control. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, pages 21847–21854. IEEE, 2026.

[11] Z. Fu, T. Z. Zhao, and C. Finn. Mobile aloha: Learning bimanual mobile manipulation using low-cost whole-body teleoperation. In *Conference on Robot Learning*, pages 4066–4083. PMLR, 2025.

[12] L. Y. Zhu, P. Kuppili, R. Punamiya, P. Aphiwetsa, D. Patel, S. Kareer, S. Ha, and D. Xu. Emma: Scaling mobile manipulation via egocentric human data. *IEEE Robotics and Automation Letters*, 2026.

[13] Y. Jiang, R. Zhang, J. Wong, C. Wang, Y. Ze, H. Yin, C. Gokmen, S. Song, J. Wu, and L. Fei-Fei. Behavior robot suite: Streamlining real-world whole-body manipulation for everyday household activities. In *Conference on Robot Learning*, pages 1246–1281. PMLR, 2025.

[14] X. Xu, J. Park, H. Zhang, E. Cousineau, A. Bhat, J. Barreiros, D. Wang, J. Bohg, and S. Song. HoMMI: Learning Whole-Body Mobile Manipulation from Human Demonstrations. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2026. doi: 10.15607/RSS.2026.XXII.205.

[15] L. Righetti, M. Kalakrishnan, P. Pastor, J. Binney, J. Kelly, R. C. Voorhies, G. S. Sukhatme, and S. Schaal. An autonomous manipulation system based on force control and optimization. *Autonomous Robots*, 36(1):11–30, 2014.

[16] Y. Huang, J. Silvério, L. Rozo, and D. G. Caldwell. Generalized task-parameterized skill learning. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 5667–5674. IEEE, 2018.

[17] H. Zhao, D. Wang, Y. Zhu, X. Zhu, O. L. Howell, L. Zhao, Y. Qian, R. Walters, and R. Platt. Hierarchical equivariant policy via frame transfer. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=nAv5ketrHq.

[18] H. Zhao, Y. Qi, B. Hu, Y. Zhu, Z. Chen, H. Tian, X. Zhu, O. Howell, H. Huang, R. Walters, et al. Generalizable hierarchical skill learning via object-centric representation. *IEEE Robotics and Automation Letters*, 11(6):6536–6543, 2026.

[19] J. Gao, X. Jin, F. Krebs, N. Jaquier, and T. Asfour. Bi-kvil: Keypoints-based visual imitation learning of bimanual manipulation tasks. In *2024 IEEE international conference on robotics and automation (ICRA)*, pages 16850–16857. IEEE, 2024.

[20] A. Bahety, P. Mandikal, B. Abbatematteo, and R. Martín-Martín. ScrewMimic: Bimanual Imitation from Human Videos with Screw Space Projection. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi:10.15607/RSS.2024.XX.078.

[21] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[22] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

[23] O. Celik, D. Zhou, G. Li, P. Becker, and G. Neumann. Specializing versatile skill libraries using local mixture of experts. In *Conference on Robot Learning*, pages 1423–1433. PMLR, 2022.

[24] O. Celik, A. Taranovic, and G. Neumann. Acquiring diverse skills using curriculum reinforcement learning with mixture of experts. In *International Conference on Machine Learning*, pages 5907–5933. PMLR, 2024.

[25] Y. Wang, Y. Zhang, M. Huo, T. Tian, X. Zhang, Y. Xie, C. Xu, P. Ji, W. Zhan, M. Ding, et al. Sparse diffusion policy: A sparse, reusable, and flexible policy for robot learning. In *Conference on Robot Learning*, pages 649–665. PMLR, 2025.

[26] K. Guo, H. Liu, Y. Sun, R. Zhao, J. Zhou, and J. Ma. Moe-act: Scaling multi-task bimanual manipulation with sparse task-conditioned mixture-of-experts transformers. *arXiv preprint arXiv:2603.15265*, 2026.

[27] C. Hao, X. Zhai, Y. Liu, and H. Soh. Abstracting robot manipulation skills via mixture-of-experts diffusion policies. In *The Fourteenth International Conference on Learning Representations*, 2026.

[28] A. Rodriguez, C. Li, L. Mazza, R. Younis, O. Hellig, S. Bodenstedt, M. Wagner, and S. Speidel. Lar-moe: Latent-aligned routing for mixture of experts in robotic imitation learning. *arXiv preprint arXiv:2603.08476*, 2026.

[29] B. Cheng, T. Liang, S. Huang, M. Shao, F. Zhang, B. Xu, Z. Xue, and H. Xu. Moe-dp: An moe-enhanced diffusion policy for robust long-horizon robotic manipulation with skill decomposition and failure recovery. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9186–9193. IEEE, 2026.

[30] M. Reuss, J. Pari, P. Agrawal, and R. Lioutikov. Efficient diffusion transformer policies with mixture of expert denoisers for multitask learning. In *International Conference on Learning Representations*, volume 2025, pages 17247–17275, 2025.

[31] J. Yu, H. Liu, Q. Yu, J. Ren, C. Hao, H. Ding, G. Huang, G. Huang, Y. Song, P. Cai, et al. Forcevla: Enhancing vla models with a force-aware moe for contact-rich manipulation. *Advances in Neural Information Processing Systems*, 38:93409–93439, 2025.

[32] D. Wang, S. Hart, D. Surovik, T. Kelestemur, H. Huang, H. Zhao, M. Yeatman, J. Wang, R. Walters, and R. Platt. Equivariant diffusion policy. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=wD2kUVLT1g.

[33] X. Xu, D. Bauer, and S. Song. RoboPanoptes: The All-Seeing Robot with Whole-body Dexterity. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025. doi:10.15607/RSS.2025.XXI.042.

[34] D. Wang, B. Hu, S. Song, R. Walters, and R. Platt. A practical guide for incorporating symmetry in diffusion policy. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=e0Dn7dg395.

[35] X. Xu, Y. Hou, Z. Liu, and S. Song. Compliant residual dagger: Improving real-world contact-rich manipulation with human corrections. *Advances in Neural Information Processing Systems*, 38:139559–139581, 2025.

[36] N. Chernyadev, N. Backshall, X. Ma, Y. Lu, Y. Seo, and S. James. Bigym: A demo-driven mobile bi-manual manipulation benchmark. In *Conference on Robot Learning*, pages 4201–4217. PMLR, 2025.

[37] Z. Jiang, Y. Xie, K. Lin, Z. Xu, W. Wan, A. Mandlekar, L. J. Fan, and Y. Zhu. Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16923–16930. IEEE, 2025.

[38] C. Kapfer, K. Stine, B. Narasimhan, C. Mentzel, and E. Candès. Marlowe: Stanford's gpu-based computational instrument, 2025. URL https://doi.org/10.5281/zenodo.14751899.

[39] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

A Implementation Details

A.1 Expert Action Parameterizations

We instantiate MoF with four experts that capture complementary action structures in bimanual mobile manipulation: left, right, base-relative (base), and per-arm relative-trajectory (rel traj). The left and right experts use the corresponding end-effector frame, F_l or F_r, at the most recent observation step. The base-relative expert F_b represents rotations and translation directions in the mobile-base frame, while re-centering each arm's translation at its observation-step end-effector position. This preserves body- and gravity-aligned structure while reducing absolute translation variance. The relative-trajectory expert F_{rt}, in contrast, represents each arm's future motion in that arm's own current end-effector frame, removing both global translation and global orientation. Thus, the base-relative expert keeps motions aligned with the robot body, while the relative-trajectory expert makes local end-effector-centric motions more compact.

A.2 MoF-MoE and MoF-Ensemble

We evaluate two variants of MoF. MoF-MoE, shown in Fig. 2, uses a router conditioned on the diffuser-conditioning embedding to predict the expert weights w_m. MoF-Ensemble removes the learned router and uses uniform weights, $w_m = 1/|\mathcal{F}|$, effectively averaging the aligned predictions from all frame experts.

A.3 Per-Expert Auxiliary Loss

The mixed objective $\mathcal{L}_{\text{mix}}$ supervises only the fused canonical-frame prediction, so experts with small router weights may receive weak gradients. We therefore add a per-expert auxiliary denoising loss that directly supervises each expert in its own frame. The final training objective is $\mathcal{L} = \mathcal{L}_{\text{mix}} + \lambda_{\text{aux}}\mathcal{L}_{\text{aux}}$. The full objective is given in Appendix B.3.

B Additional Method Details

B.1 Frame Transformation Operators

Here we provide the explicit transformation operators used in Eq. (1). Let ${}^mT_c = ({}^mR_c, {}^mt_c)$ denote the rigid transform from the canonical frame F_c to an expert frame F_m. For one arm at one action timestep, the canonical-frame action channels are $x_c = [p_c, c_c^1, c_c^2, g]$, where p_c is the end-effector position, c_c^1, c_c^2 are the first two rotation columns, and g is the gripper command.

The action-state transformation ${}^m\mathcal{T}_c$ is applied to noisy action states. For each arm and timestep, it maps $p_c \mapsto {}^mR_c p_c + {}^mt_c$, $c_c^i \mapsto {}^mR_c c_c^i$, and $g \mapsto g$. Equivalently, for the vectorized action chunk, ${}^m\mathcal{T}_c(x_c) = {}^mA_c x_c + {}^mb_c$, where ${}^mA_c$ applies ${}^mR_c$ to every position and rotation-column channel and leaves gripper channels unchanged, while ${}^mb_c$ inserts the translation ${}^mt_c$ only into the position channels.

The denoiser output has the same channel layout, $\epsilon_c = [\Delta p_c, \Delta c_c^1, \Delta c_c^2, \Delta g]$, but represents a noise vector in the diffusion action space rather than an absolute pose. Therefore, we use the associated linear operator ${}^m\mathbf{T}_c$, defined as ${}^m\mathbf{T}_c(\epsilon_c) = {}^mA_c\epsilon_c$. Equivalently, $\Delta p_c \mapsto {}^mR_c\Delta p_c$, $\Delta c_c^i \mapsto {}^mR_c\Delta c_c^i$, and $\Delta g \mapsto \Delta g$. The inverse operators ${}^c\mathcal{T}_m$ and ${}^c\mathbf{T}_m$ are defined analogously using the inverse rigid transform ${}^cT_m = ({}^mT_c)^{-1}$.

B.2 Frame-specific Conditioning

All experts share the same visual features, since RGB observations are not tied to a particular action frame. In contrast, proprioceptive inputs such as end-effector poses and gripper states are transformed into each expert's action parameterization before being provided to that expert. The base-relative and relative-trajectory experts share the same base-frame proprioceptive features.

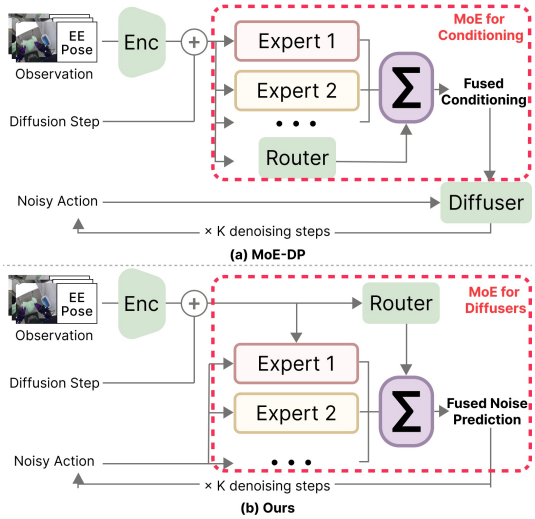


Figure 7: **Difference Between MoE-DP and Our Method.** (a) MoE-DP introduces mixture-of-experts structure in the conditioning pathway, but the diffusion process still denoises actions in a single action representation. (b) MoF instead places the mixture structure directly over frame-specialized denoisers: each expert denoises the same underlying action state expressed in a different coordinate frame. Thus, MoF performs mixture-of-experts reasoning over action frames rather than over conditioning features.

B.3 Per-Expert Auxiliary Loss

The mixed objective $\mathcal{L}_{\text{mix}}$ supervises only the fused canonical-frame prediction. When the router assigns a small weight to an expert, that expert may receive weak gradients from the mixed loss and drift away from being a meaningful denoiser in its own frame. To stabilize training, we add a per-expert auxiliary loss that directly supervises each expert against the diffusion noise expressed in its own frame.

Given a canonical-frame noisy action x_c^k and canonical-frame diffusion noise ϵ_c, expert m receives ${}^m\mathcal{T}_c(x_c^k)$ and predicts noise in frame F_m. The corresponding target is the transformed noise ${}^m\mathbf{T}_c(\epsilon_c)$. The auxiliary loss is

$$\mathcal{L}_{\text{aux}}(\theta) = \mathbb{E}_{k,\epsilon_c,o}\left[\sum_{F_m \in \mathcal{F}} \left\| {}^m\mathbf{T}_c(\epsilon_c) - \epsilon_m^\theta({}^m\mathcal{T}_c(x_c^k), o; k) \right\|^2\right].$$

The final training objective is $\mathcal{L} = \mathcal{L}_{\text{mix}} + \lambda_{\text{aux}}\mathcal{L}_{\text{aux}}$. This auxiliary term is used only during training; inference uses the fused noise prediction in Eq. (1).

B.4 Difference Between MoE-DP and MoF

Fig. 7 illustrates the difference between a standard mixture-of-experts diffusion-policy baseline and our multi-frame denoising formulation. In MoE-DP, the mixture structure is applied to the conditioning pathway: multiple experts process the observation or conditioning features, their outputs are fused, and a single diffuser denoises actions in one fixed action representation. Therefore, although MoE-DP increases model capacity and allows input-dependent conditioning, the diffusion process itself still operates in a single coordinate frame.

MoF instead places the mixture structure directly over the denoising process. At each diffusion step, the same canonical noisy action is re-expressed in multiple expert frames, each expert predicts noise in its own action representation, and the predicted noises are transformed back to the canonical frame before fusion. Thus, the experts in MoF are not merely alternative conditioning modules; they are frame-specialized denoisers that operate on different coordinate-frame parameterizations of the same underlying action state. This distinction is important for our experiments: the comparison against

MoE-DP tests whether the gains come from synchronized multi-frame action denoising rather than from adding a generic MoE module.

C Simulation Details

This section provides the full implementation details for our simulation experiments.

C.1 Benchmarks and Demonstration Data

We evaluate on nine bimanual tasks spanning two benchmarks. Five tasks use the BiGym [36] mobile-manipulation benchmark with the RBY1 bimanual mobile manipulator (*Flip Cup*, *Move 2 Plates*, *Kitchenware*, *Flip Sandwich*, *Dishwasher*), and four tasks use the DexMimicGen [37] tabletop benchmark (*Threading*, *3 Piece Assembly*, *Box Cleanup*, *Drawer Cleanup*).

For both benchmarks, we generate training data by implementing the DexMimicGen [37] data-generation algorithm and extending it to the mobile-manipulation setting when needed. We collect 100 demonstrations per task for all nine tasks. We follow the task-specific camera configuration defined in each benchmark: two wrist cameras combined with an egocentric head view for the BiGym RBY1 tasks, and two wrist cameras combined with a third-person view for the DexMimicGen tasks.

C.2 Observation and Action Space

All policies share the same observation space and use a common action layout that differs only in the gripper/hand command. The visual observation consists of three RGB camera images (two wrist views and one egocentric/third-person view), each rendered at 84×84. All images are preprocessed to 76×76 before being passed to the vision encoder; during training we apply a random crop, and at test time a center crop. The low-dimensional observation consists of the two end-effector poses, the proprioceptive gripper/hand state, and the base/head pose; these proprioceptive inputs are re-expressed in each expert's action parameterization before being given to that expert. We use an observation horizon of 2 steps.

Each arm contributes a 3D end-effector position and a 6D rotation representation (the first two columns of the rotation matrix). The five BiGym tasks and the two parallel-jaw DexMimicGen tasks (*Threading*, *3 Piece Assembly*) use a single parallel-jaw gripper per arm, giving a 20-dimensional action (9 per arm plus one gripper command per arm) and a 2-dimensional gripper proprioception. The two dexterous-hand DexMimicGen tasks (*Box Cleanup*, *Drawer Cleanup*) instead use a 6-DoF multi-finger hand per arm, giving a 30-dimensional action (9 pose channels plus 6 finger-joint commands per arm) and a 24-dimensional hand proprioception (12 joints per hand). The MoF frame transformations act only on the per-arm position and rotation channels and leave the gripper/finger channels unchanged, so the same multi-frame denoising procedure applies to both the 20- and 30-dimensional actions. Policies predict an action chunk of horizon 16 and execute the first 8 predicted actions before re-planning.

C.3 Training Hyperparameters

All policies (MoF and all baselines) share the same diffusion backbone, vision encoder, optimizer, and training schedule; they differ only in the action parameterization and in the method-specific modules described below.

We use the standard Diffusion Policy [1] conditional 1D U-Net as the denoiser, conditioned on the observation embedding through FiLM. The U-Net uses channel dimensions $[256, 512, 1024]$, a diffusion-step embedding of dimension 128, a convolution kernel size of 5, and 8 GroupNorm groups. The vision encoder is a ResNet-18, pretrained on ImageNet and fine-tuned end-to-end, with spatial-softmax feature aggregation and GroupNorm; we do not share the encoder across the three cameras. Images are preprocessed to 76×76, with a random crop during training and a center crop at test time. We use an observation horizon of 2 steps, predict an action chunk of horizon 16, and

execute the first 8 actions before re-planning. The diffusion process uses a DDIM [39] scheduler with 50 training timesteps and 16 inference timesteps.

We optimize all policies with AdamW using a learning rate of 1×10^{-4}, weight decay of 1×10^{-6}, and $(\beta_1, \beta_2) = (0.95, 0.999)$. The learning rate follows a cosine schedule with 500 warmup steps. We train with a batch size of 128 for 500 epochs.

C.4 MoF Hyperparameters

MoF augments the shared backbone with four frame experts and a learned router (for MoF-MoE). The expert set is {base-relative, left, right, rel-traj}, and the base-relative frame is used as the canonical frame F_c for the synchronized denoising trajectory. MoF-MoE uses an MLP router with hidden dimension 512 that takes the diffuser-conditioning embedding together with the diffusion-step embedding and outputs a softmax distribution over the four experts. MoF-Ensemble replaces the router with fixed uniform weights $w_m = 1/|\mathcal{F}| = 1/4$. Both variants use the per-expert auxiliary denoising loss of Appendix B.3 with weight $\lambda_{\text{aux}} = 1$.

C.5 Evaluation Protocol

Each policy is trained for 500 epochs with three random seeds. For each seed, we evaluate five checkpoints from epochs 460–500 at 10-epoch intervals, using 50 rollout episodes per checkpoint. We report the mean success rate over the five checkpoints and three seeds, together with the standard error across the three seeds. Rollout initial configurations use a held-out seed range disjoint from the training demonstrations.

D Router Analysis

We analyze the learned MoF-MoE router across all nine simulation tasks. For each task we load the trained MoF-MoE policy and run it open-loop on 20 training demonstrations, recording the router's four-expert weight vector at every observation window and every reverse-diffusion step. Throughout this section we focus on the *late denoising phase*, i.e., the per-window weight vector averaged over the last 4 of the 16 reverse-diffusion steps. We restrict attention to these clean steps because the router is essentially observation-independent during the high-noise steps (where the action is close to Gaussian noise and provides little signal), and only develops state-dependent structure as the action sharpens.

D.1 Within-Episode Router Trajectories for All Tasks

Fig. 8 shows the router-weight trajectories for all nine tasks, overlaid with head-camera observations at sampled task progressions. The trajectories vary substantially across tasks in both their static frame preference and their within-episode dynamics. *Threading* and *Drawer Cleanup* show the clearest within-episode handoff: the rel-traj expert dominates the early phase, where each arm independently reaches and grasps its object, and the base expert takes over for the later contact-rich phase that requires coordinated bimanual motion. *Flip Sandwich* and *Kitchenware* keep the rel-traj expert on top throughout with only mild drift, while *3 Piece Assembly*, *Dishwasher*, and *Move 2 Plates* retain a base-dominated soft mixture. *Box Cleanup* is an outlier: all four expert weights cluster tightly near 0.25 across all phases, indicating a near-uniform router that does not specialize to any frame or phase. *Flip Cup* is notable in the opposite way: the relative ordering of experts is inverted from the other tasks, with the rel-traj expert always on top and the base expert always at the bottom.

Reading the router trajectories together with the observation frames suggests that the within-episode router dynamics track the visible phase structure of each task. On the tasks with the largest within-episode modulation, the shift from the rel-traj to the base expert coincides with the transition from independent, arm-centric reaching, which is most compact in each arm's own trajectory frame, to coordinated bimanual manipulation, which is most naturally expressed in the shared body frame. This

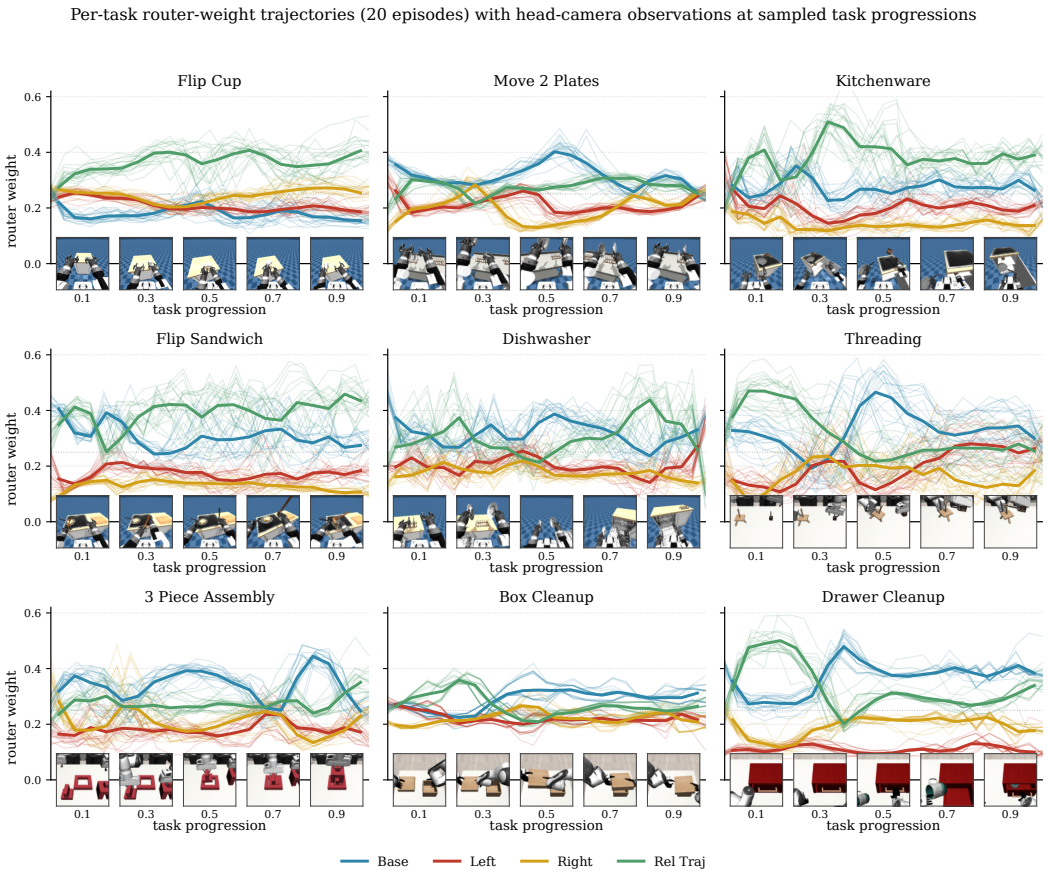


Figure 8: **Router weight trajectories for all nine tasks (20 demonstrations each), with camera observations at sampled task progressions.** For each task, faint lines show the per-episode router weight for each of the four experts (averaged over the last 4 of 16 denoising steps) versus task progression, and bold lines show the across-episode mean; the dotted line at 0.25 marks the uniform-weight reference. Below each panel, camera observations from the demonstration whose router trajectory is closest to the across-episode mean are shown at task progressions 0.1, 0.3, 0.5, 0.7, 0.9.

phase-aligned routing is most pronounced precisely on the tasks where MoF-MoE most improves over Oracle Frame (Appendix D.2), and is essentially absent on *Box Cleanup*, whose near-uniform router cannot exploit such structure. Taken together, this indicates that the learned router contributes most when a task progresses through phases that favor different action frames.

D.2 Router Modulation Amplitude vs. MoF-MoE Advantage over Oracle Frame

To quantify the within-episode modulation observed in Fig. 8, we summarize each task by a single scalar that we call the *router modulation amplitude*. We partition each episode's windows into 10 within-episode time bins, compute the mean late-phase weight per expert per bin, take the standard deviation of each expert's binned-mean weight across the 10 bins, and average over the four experts. Intuitively, this scalar measures how much each expert's weight typically swings over the course of an episode.

Fig. 9 plots the router modulation amplitude against MoF-MoE's per-task advantage over Oracle Frame. The two quantities are positively correlated across tasks (Spearman $\rho = +0.75$, $p = 0.020$, $n = 9$). The two tasks with the largest router modulation, *Threading* and *Drawer Cleanup*, are also the tasks where MoF-MoE most improves over the best single frame ($+7.7$ and $+6.8$ points). At the other end of the spectrum, *Box Cleanup* has a relatively low router modulation amplitude—its router

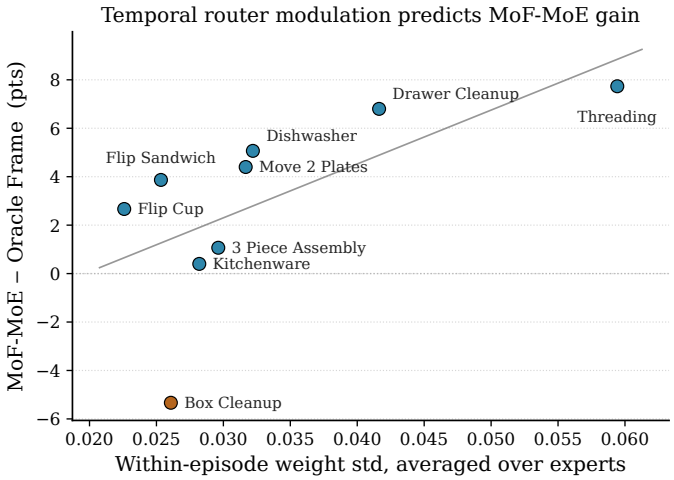


Figure 9: **Router modulation amplitude vs. MoF-MoE advantage over Oracle Frame.** Each point is one task. The x-axis is the within-episode standard deviation of each expert's binned-mean weight, averaged over the four experts. The y-axis is the difference between MoF-MoE's and Oracle Frame's last-5 success rate. The orange marker (*Box Cleanup*) is the only task where MoF-MoE underperforms Oracle Frame and is also the task with relatively low router modulation amplitude. Spearman $\rho = +0.75$, $p = 0.020$, $n = 9$.

is close to uniform throughout each episode—and is the only task where MoF-MoE underperforms Oracle Frame. *Move 2 Plates*, *Dishwasher*, *Flip Sandwich*, and *Flip Cup* show smaller gains over Oracle Frame than Threading and Drawer Cleanup. This relationship suggests that MoF-MoE's advantage over the best single frame stems primarily from its ability to assign different frames to different task phases, rather than from a static preference for any one frame.

E Ablation Study Details

We provide the full ablation results for the four representative tasks used in Sec. 4: *FlipCup*, *Move2Plates*, *Threading*, and *BoxCleanup*. These tasks cover both BiGym and DexMimicGen environments and include settings where different action parameterizations are favored. Parenthetical values denote differences from the corresponding baseline: MoF-MoE for MoE variants and MoF-Ensemble for the ensemble variant.

Table 4: Full ablation study for MoF. Last-5 success rate (%, mean ± std. err. over 3 seeds). Parenthetical values denote differences from the corresponding baseline.

Method	Avg	Flip Cup	Move 2 Plates	Threading	Box Cleanup
MoF MoE (Ours)	66.5	56.5 ± 1.4	51.6 ± 4.4	70.8 ± 7.5	87.1 ± 3.2
Rel Traj as Canonical	64.0 (−2.5)	56.8 ± 3.6 (+0.3)	45.9 ± 3.3 (−5.7)	67.6 ± 2.6 (−3.2)	85.7 ± 5.3 (−1.4)
Left as Canonical	65.8 (−0.7)	56.8 ± 2.0 (+0.3)	51.3 ± 3.8 (−0.3)	67.6 ± 1.2 (−3.2)	87.6 ± 6.2 (+0.5)
Right as Canonical	66.4 (−0.1)	55.3 ± 2.6 (−1.2)	54.7 ± 6.1 (+3.1)	71.3 ± 2.2 (+0.5)	84.3 ± 1.4 (−2.8)
w/o Best Frame	60.3 (−6.2)	54.9 ± 2.9 (−1.6)	45.1 ± 5.3 (−6.5)	63.2 ± 5.0 (−7.6)	77.9 ± 8.6 (−9.2)
w/o Aux	58.5 (−8.0)	45.5 ± 2.9 (−11.0)	30.8 ± 0.9 (−20.8)	70.3 ± 6.1 (−0.5)	87.5 ± 5.3 (+0.4)
w/ Ortho Proj	59.5 (−7.0)	49.2 ± 2.1 (−7.3)	38.4 ± 3.8 (−13.2)	64.4 ± 8.9 (−6.4)	85.9 ± 1.4 (−1.2)
MoF Ensemble (Ours)	64.2	59.2 ± 0.4	51.5 ± 1.5	68.7 ± 3.3	77.5 ± 5.2
w/ Ortho Proj	0.0 (−64.2)	0.0 ± 0.0 (−59.2)	0.0 ± 0.0 (−51.5)	0.0 ± 0.0 (−68.7)	0.0 ± 0.0 (−77.5)

Changing the canonical frame has only a modest effect. Using relative-trajectory, left, or right as the canonical frame changes the average success rate by at most 2.5 percentage points, suggesting that MoF is not sensitive to the arbitrary choice of scheduler frame. Removing the best-performing single frame from the expert set reduces the average success rate by 6.2 percentage points, showing that

Table 5: Expert frame scalability: last-5 success rate (%, mean ± std. err. over 3 seeds). Bold indicates best.

Frame set	Flip Cup	Move 2 Plates	Kitchenware	Flip Sandwich
1F DP (oracle frame)	53.9 ± 2.9	47.2 ± 1.7	31.2 ± 2.6	36.1 ± 3.3
3F MoE (w/o oracle frame)	54.9 ± 2.9	45.1 ± 5.3	34.5 ± 1.3	37.7 ± 2.9
4F MoE (Ours)	56.5 ± 1.4	51.6 ± 4.4	31.6 ± 2.4	$\mathbf{40.0 \pm 3.1}$
4F + mid-torso-relative	55.7 ± 5.0	47.5 ± 2.3	32.9 ± 2.2	36.8 ± 0.6
4F + egoview-relative	57.7 ± 1.9	53.2 ± 1.5	28.4 ± 2.8	39.5 ± 2.4
4F + shoulder-relative	$\mathbf{58.0 \pm 2.3}$	46.4 ± 2.7	28.9 ± 4.1	36.8 ± 1.4
4F + elbow-relative	56.9 ± 1.1	44.1 ± 2.3	29.3 ± 2.2	33.9 ± 2.1
4F + EE-midpoint-relative (virtual)	55.6 ± 0.9	$\mathbf{53.7 \pm 3.0}$	$\mathbf{34.8 \pm 1.3}$	32.1 ± 1.9

the expert set is not redundant and that task-relevant frames contribute complementary denoising structure.

The orthogonalization ablation validates the transformation-compatible action representation. When noisy rotation channels are projected back to $\mathrm{SO}(3)$ before applying frame transformations, MoF-MoE drops by 7.0 percentage points on average, and MoF-Ensemble collapses completely. This is because the orthogonalization completely destroys all non-canonical-frame experts. Although MoF-MoE is able to escape by assigning all weights to the canonical expert, there is no way for MoF-Ensemble to recover from this failure. This supports our design choice of using column-vector rotation channels, which can be transformed directly as ordinary 3D vectors without changing the noisy diffusion state.

Finally, removing the per-expert auxiliary denoising loss reduces average performance by 8.0 percentage points. This suggests that directly supervising each expert in its own frame is important for keeping all frame experts meaningful during training, especially when the fused loss may provide weak gradients to experts with small router weights.

F Expert Frame Selection and Scalability

We investigate how the number and choice of expert frames affect MoF performance and computational cost. We evaluate MoF-MoE on four BiGym tasks (*Flip Cup*, *Move 2 Plates*, *Kitchenware*, and *Flip Sandwich*), comparing the best single-frame policy, a three-expert variant excluding the oracle frame, and our default four-expert configuration. We further augment the default expert set with five additional frames: mid-torso, egoview, shoulder, elbow, and a virtual end-effector midpoint. We also measure training and inference latency and GPU memory usage as the number of experts increases.

Table 5 shows that removing the oracle frame degrades performance on average, while adding a fifth frame improves some tasks but provides no consistent gains across tasks. Computational costs scale approximately linearly with the number of experts, with training and inference latency increasing by 22.1 and 58.5 ms per expert, and GPU memory by 1.94 and 0.33 GiB, respectively, on an RTX PRO 6000 Blackwell. Thus, adding more frames incurs additional computational cost without guaranteed performance gains. Our default four-frame configuration provides a natural, compact set of complementary reference frames for bimanual mobile manipulation without requiring task-specific or embodiment-specific frame design.

G Real-World Experiment Details and Analysis

G.1 Task Definitions

We evaluate two real-world tasks: pouring and serving. In the **pouring task**, the robot picks up two cups, transfers a red cube from the right cup to the left cup, and places both cups upright on the table. Success requires the cube to end in the left cup and the robot to release both cups while they remain

upright. The initial configurations vary the ordered right/left cup-color pair and the table placement distance from the robot over four distance levels. The seen color pairs are (Yellow, Gray), (Yellow, Blue), and (Gray, Blue), while the unseen color pairs are (Blue, Purple) and (Purple, Gray). In the **serving task**, the robot navigates toward the table, picks up a cup, and places it upright on the table. Success requires the cup to be placed upright without falling or being released incorrectly. We vary the plate color and the distance between the robot's initial base pose and the table. Purple plates are seen during training and are evaluated at six base-to-table distances, while Blue and Milk plates are unseen and are evaluated at seven distances, including one additional closest-distance setting.

G.2 Training Details

We integrate our MoF policy within the HoMMI [14] framework by implementing our column-vector SE(3) action representation and frame transformations, while keeping the HoMMI runtime interface unchanged. The MoF-MoE policy uses left, right, and relative-trajectory experts, with the left frame as the canonical frame. Since the HoMMI data collection interface does not record the base frame, we disable the base frame and exclude the base-relative expert. For the relative-trajectory expert, low-dimensional observations are therefore expressed using the left-frame observation variant. The head look-at action is represented as a 3D point rather than a pose, so it contains no orientation channel and is transformed using point-frame transformations. For the relative-trajectory expert, this look-at point is also expressed in the left frame. We train real-world MoF-MoE with the same objective as in simulation, including the mixed canonical denoising loss and the per-expert auxiliary denoising loss.

All evaluated policies use a DiT-based diffusion-policy backbone. The policies use an observation horizon of 2, an action horizon of 32, represent rotations with 6D rotation channels, and predict a 23-dimensional action vector. The action vector consists of two end-effector target poses, each represented by a 3D position and a 6D rotation, two gripper commands, and a 3D head look-at point. The DDIM scheduler uses 100 training timesteps, a squared-cosine beta schedule, and epsilon prediction. During inference, we use 16 denoising steps. Visual preprocessing follows the training pipeline. Head images are resized to 224×224, wrist images are cropped and resized to 96×96, and the pointmap branch samples 512 points from the head pointmap. Training augmentation applies color jitter to the head stream. For wrist streams, we apply center crop, random crop, color jitter, and resize. We train with a policy learning rate of 7.5×10^{-5}, a visual-encoder learning rate of 7.5×10^{-6}, weight decay of 10^{-6}, a cosine learning-rate schedule, 50 warmup steps, EMA, batch size 16, and gradient accumulation of 2. We evaluate the epoch-500 checkpoints for serving and the epoch-580 checkpoints for pouring.

G.3 Deployment Details

We adopt the HoMMI [14] hardware setup and whole-body controller for all real-world rollouts. The policy server runs at 5 Hz with an action horizon of 32. The controller executes policy trajectories at 5 Hz, while state estimation and IK run at 100 Hz. A new policy segment is requested every 3.6 s for serving and every 3.2 s for pouring. The controller linearly interpolates the received policy targets and applies a 1 Hz low-pass filter to the commands. For safety and compliance during real-world execution, we enable both the compliant controller and the end-effector wrench safety guard. The admittance controller uses zero desired wrench, translational stiffness $(300, 300, 300)$, rotational stiffness $(1, 1, 1)$, translational damping $(7, 7, 7)$, and rotational damping $(0.2, 0.2, 0.2)$. We use translation-force mode with direct force-control gains set to zero. The wrench safety guard uses calibrated end-effector wrench readings and latches the output when the measured force exceeds $120\ \mathrm{N}$ or the measured torque exceeds $12\ \mathrm{Nm}$.

The same whole-body IK parameters are used for both serving and pouring. The interpolated Cartesian targets are converted into generalized joint velocities using the HoMMI differential whole-body IK solver. We use the `daqp` optimizer with a damping coefficient of 10^{-6} for numerical stability. The IK objective prioritizes accurate bimanual end-effector tracking, with equal position and orientation

tracking weights of 10000. We regularize the robot toward a nominal torso posture with weight 200, while disabling nominal arm-posture regularization. To smooth the solution in velocity space, we penalize deviations from the current configuration with weight 10 for the joints. The mobile base is regularized more strongly, with both translational and rotational base components weighted by 5000. Unlike the original HoMMI controller, we disable the center-of-mass-over-base objective, while keeping the relative torso–base displacement bounds at $0.15\ \mathrm{m}$ in both the x and y directions.

G.4 Failure Mode Analysis

In the pouring task, **MoF-MoE** achieves 85% success, with only three failures: two right-cup pickup failures and one left-cup pickup failure, all on seen cup-color combinations. Importantly, in all 17 rollouts where MoF-MoE successfully picks up both cups, it completes both cube transfer and upright placement, suggesting that multi-frame fusion primarily improves robustness in the initial grasping phase while preserving reliable bimanual coordination afterward. The single-frame baselines fail in distinct, frame-dependent ways. The **Left** policy achieves only 15% success, with 12 right-cup pickup failures, two left-cup pickup failures, two pouring failures, and one rollout that repeatedly continues the pouring motion even after successful transfer. It fails on all unseen color combinations, and its errors are dominated by the right arm. Conversely, the **Right** policy achieves 70% success, but all of its failures are left-side failures: four left-cup pickup failures and two left-cup placement failures, all on seen color combinations. These left- and right-frame results suggest that when the policy is trained in a single end-effector frame, predictions for the opposite arm become insufficiently precise. The **Rel Traj** policy achieves only 5% success and is highly unstable during precise pickup, with 13 left-cup pickup failures, three failures to pick up both cups, two transfer failures due to insufficient bimanual coordination, and one right-cup placement failure. This indicates that relative-trajectory actions alone are not stable enough for the fine-grained grasping required by pouring.

In the serving task, **MoF-MoE** achieves 70% success, with four cup-pickup failures and two placement failures. One failure occurs on seen plate-color configurations and five occur on unseen plate-color configurations. The failures are concentrated at the closest and intermediate base-to-table distances, while MoF-MoE succeeds on all configurations that require longer-distance navigation, indicating stable navigation over extended approach motions. The **Left** policy achieves 55% success and fails nine times at cup pickup. More than half of these failures occur at farther-than-average base-to-table distances, suggesting that long-distance navigation causes the cup to shift on the plate and leads to subsequent pickup failure. The **Right** policy fails in all 20 rollouts and exhibits highly unstable navigation throughout, suggesting that using the right frame as the sole reference induces a wider and harder-to-learn action distribution for this task. The **Rel Traj** policy achieves 60% success, with all eight failures occurring during the cup-pickup phase due to insufficient bimanual coordination. In contrast to the Left policy, more than half of the relative-trajectory failures occur at closer-than-average base-to-table distances, indicating that relative-trajectory actions support stable navigation but do not provide the coordinated bimanual reference needed for grasping the cup from the plate.