Frequency-Aware Flow Matching for Continuous and Consistent Robotic Action Generation

Jianing Guo[1, 4], Fangzheng Chen[1, 4], Zihao Mao[1], Wong Lik Hang Kenny[3], Zhenhong Wu[1], Yu Li[2, 4], Yishuai Cai[4], Yuanpei Chen[2, 4], Yikun Ban[1], Kai Chen[3], Qi Dou[3], Xianglong Liu[1, 5, 6], Huijie Zhao[1], Simin Li[1, 3 *], Yaodong Yang[2, 4]

[1]Beihang University, [2]Peking University, [3]The Chinese University of Hong Kong, [4]PKU-Psibot Lab, [5]Zhongguancun Laboratory, [6]Hefei Comprehensive National Science Center

Abstract

Flow matching has emerged as a standard paradigm for robotic manipulation owing to its strong expressive power for modelling complex, multimodal action distributions, alongside similar approaches like diffusion policy. However, existing methods rely on discretized action chunks, making them brittle to demonstrations collected at heterogeneous control frequencies and prone to temporally inconsistent actions that degrade control stability. In this paper, we propose Frequency-Aware Flow Matching (FAFM), which outputs continuous, temporally consistent actions. To handle heterogeneous frequency input, we transform discrete action sequences into the frequency domain with the discrete cosine transform (DCT), perform flow matching over the resulting coefficients, and reconstruct continuous actions via cosine basis expansion. To generate temporally consistent actions, we regularize the first-order temporal derivative to promote smooth actions. This corresponds to a Sobolev-type constraint that suppresses high-frequency errors and discourages abrupt action changes. Our FAFM is simple, introduces no additional network parameters and applies to standalone flow-matching policies and vision-language action models. Across synthetic toy benchmark, obstacle avoidance, LapGym, and LIBERO, FAFM improves success rates, multimodal expressivity, motion smoothness, convergence speed, robustness to mechanical bias and mixed-frequency input. These gains are consistent when deployed on a real-world Franka robot. Code available at `https://github.com/gakakulicc/FAFM`.

1 Introduction

Flow matching [1, 2] endows robot policies to represent complex, multimodal action distributions. Owing to this expressive power, flow matching has been adopted as the action head in several robot foundation models [3, 4, 5, 6, 7, 8], enabling flexible, generalizable, and dexterous manipulation in real-world settings. To support real-time inference, such policies typically predict an action chunk [9], which is a sequence of consecutive actions at discrete time steps, executed at a fixed control frequency.

However, real-world robot motions are inherently smooth and continuous. As illustrated in Fig. 1, discretizing trajectories into action chunks introduces problems at both training and inference time. During training, the policy must reconcile demonstration data recorded at heterogeneous control frequencies that potentially cause gradient conflict. For example, the Open X-Embodiment dataset [10] spans a wide range of control frequencies, from 3 Hz (RT-1 [11]) to 50 Hz (ALOHA [9]), yet the policy is required to produce action chunks at a single fixed frequency. This discrepancy means that frequency information in the training data is largely overlooked. Consequently, the same robot executing the same physical trajectory at different frequencies will produce action chunks of different lengths and magnitudes, each of which is nonetheless a valid supervision signal.

*Corresponding Author. E-mails: lisiminsimon@buaa.edu.cn.

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

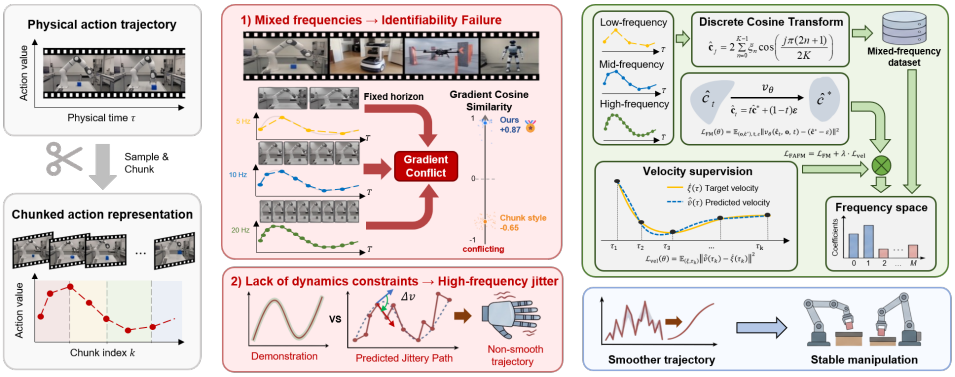

Figure 1: Introduction overview. Action chunk discretization causes Identifiability Failure and high-frequency jitter, which FAFM resolves via frequency-domain flow matching and velocity supervision.

During inference, actions at discrete timesteps are evaluated independently, without accounting for inter-timestep consistencies. Consequently, errors at adjacent timesteps may individually appear small, yet correspond to mutually conflicting motion directions. Executing such actions produces jittery, discontinuous motion that compromises control stability [12, 13, 14]. Although this may not directly reduce task success rates in rigid-body settings, it poses serious risks in soft-body manipulation. For instance, during liquid pouring, abrupt motion discontinuities can cause unintended spillage; in robotic surgery, jittery instrument motion can drive soft tissue into path-dependent deformation states that are difficult or impossible to reverse [15].

In this paper, we propose learning continuous and temporally consistent actions directly via Frequency-Aware Flow-Matching (FAFM). To accommodate discrete action trajectories collected at arbitrary sampling frequencies in training time, we transform the data into the frequency domain using the discrete cosine transform (DCT) [16]. Although DCT was recently introduced to the robotic community for real-time execution in FAST [17], we exploit it here as a basis function that naturally separates components at different frequencies. Flow matching is then performed in the frequency domain, and the predicted DCT coefficients parameterize a continuous-time action trajectory via a cosine basis expansion, which can be evaluated at arbitrary temporal resolutions. To further promote temporal consistency during inference, we additionally apply flow matching to the first-order temporal derivative of the action sequence. Because the DCT provides a continuous action representation in frequency space, this derivative is well defined and accurate. This formulation corresponds to a Sobolev-type regularization, in which errors at higher frequencies are penalized quadratically stronger than lower frequencies. Consequently, the resulting loss suppresses high-frequency errors and effectively discourages abrupt changes in the generated actions.

Overall, our FAFM is simple yet effective. It introduce no additional networks or learnable parameters, and applies to standalone flow matching policy and VLAs. Experiments on synthetic toy benchmarks, obstacle avoidance, LapGym [18], and LIBERO [19] demonstrate that our FAFM consistently achieves higher success rate than strong baselines. On synthetic toy benchmark consisting of a pair of antisymmetric sinusoidal trajectories, our approach uniquely captures the correct multimodal structure. In obstacle avoidance, our FAFM simultaneously achieves high solution diversity and strong overall performance. On LapGym, a robotic surgery benchmark that requires soft-body manipulation, our method achieves smooth control and faster convergence. On LIBERO benchmark for VLAs, our FAFM demonstrate superior motion smoothness, robustness to mechanical bias and mixed-frequency input. The superiority is consistent when deployed to real-world Franka robot.

Contribution. Our contributions are twofold. (1) We introduce FAFM, a frequency-aware flow matching method for continuous and temporally consistent action generation. FAFM applies to standalone flow matching and VLAs. (2) On various benchmarks, our FAFM achieves higher success rates, improved multimodal expressivity, faster convergence, higher motion smoothness and robustness against mechanical bias and heterogeneous input frequencies.

2 Related Work

Generative Policy Learning for Robot Action Generation. Recent advances in diffusion and flow-matching methods have reshaped generative action policies. For vision-based control, diffusion

policy [20] first introduced the paradigm of generating fixed-horizon action chunks using denoising diffusion probabilistic models (DDPMs [21, 22]), which has since become a standard approach for continuous robot control. 3D Diffusion Policy [23] extended this framework to point-cloud observations, demonstrating that explicit three-dimensional representations substantially improve generalization in dexterous manipulation tasks, with follow-ups including [24, 25, 26, 12]. More recently, flow-matching–based policies have emerged as a more inference-efficient alternative, achieving action quality comparable to diffusion methods while significantly reducing sampling latency [27]. FlowPolicy [28] further extended flow matching to 3D manipulation settings, confirming its effectiveness under complex spatial interactions, with similar works including [29, 30, 14, 31].

Parallel developments have occurred in large robot foundation models. Early systems such as RT-1 [11], RT-2 [32] and OpenVLA [33] predict discrete action tokens autoregressively and subsequently decode them into continuous control signals, trading off expressivity for scalability. More recent vision-language-action (VLA) models including Octo [34], CogACT [35], RDT-1B [36], DexGraspVLA [37] and GR00T series [38] etc have shifted toward diffusion-based action heads to improve multimodal expressivity, robustness, and cross-embodiment generalization. Flow-matching–based action generation was later adopted in π_0 [3], demonstrating both improved inference efficiency and strong control performance; this design has since been inherited by successors such as $\pi_{0.5}$ [4], $\pi^*_{0.6}$ [39], $\pi_{0.7}$ [40] , SmolVLA [5], X-VLA [6] and A1 [7], which now represents the state of the art for high-performance continuous action generation in robot foundation models.

Improving Action Representation for Robot Control. For autoregressive actions, FAST [17] applies DCT to compress high-frequency action chunks into discrete tokens, enabling efficient action execution. NIAF [41] extends the parallel decoding regression paradigm introduced by OpenVLA-OFT [42] with implicit neural representations for continuous action function regression. However, autoregressive methods have limited capability in representing multimodal actions, thus later approaches moves towards diffusion-based methods [20]. For diffusion policies, MPD [12] (and its consistency-distilled acceleration FRMD [43]) represent actions as low-dimensional probabilistic weights over temporal basis functions (ProDMP [44]) for surgical tasks, generating smooth trajectories via globally smooth basis functions. However, this global parameterization couples all local trajectory decisions into a single weight vector, smoothing away local multimodal distinctions. FreqPolicy [45] performs diffusion in a learned latent space structured by DCT frequency bands, but the learned latent entangles frequency content with observation features, causing mode leakage in multimodal settings. As for flow matching policies, SFP [46] aligns the flow matching integration time with robot execution time, streaming actions on-the-fly at each ODE step. While achieving faster policy execution and better multi-modal representation capability, SFP operates under a receding horizon scheme and does not capture the joint distribution over future actions globally. ABPolicy [14] adopts flow matching in a B-spline control-point space to enforce trajectory smoothness within and across action chunks, with asynchronous inference for real-time control.

3 Method

In this section, we propose frequency-aware flow matching to generate continuous and temporally consistent actions. Our overall approach are illustrated in Figure 2.

Problem Formulation. We consider robot action generation as learning $p(\mathbf{A}|\mathbf{o})$ over action chunks $\mathbf{A} = [a_0, \ldots, a_{K-1}] \in \mathbb{R}^{K\times d}$ given observation $\mathbf{o}$. Without loss of generality, we use the conditional flow matching paradigm used in π series [3], which is widely used in modern robot foundation models, and can be extended to similar paradigms like diffusion policy [20] easily. In conditional flow matching, given $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and target chunk $\mathbf{A}$, one trains a velocity field v_θ via:

$$\mathcal{L}_{\mathrm{FM}}(\theta) \;=\; \mathbb{E}_{(\mathbf{o},\mathbf{A}),\,t,\,\varepsilon}\big\|v_\theta(\mathbf{A}^t,\,\mathbf{o},\,t) - (\mathbf{A}-\varepsilon)\big\|^2, \qquad \mathbf{A}^t = t\mathbf{A} + (1-t)\varepsilon. \tag{1}$$

Let $p_{\mathcal{D}}$ denote the demonstration distribution over observations, trajectories $\xi : [0, T_\xi] \to \mathbb{R}^d$, and control frequencies f; define chunk length $K(\xi, f) = \lfloor T_\xi f \rfloor$ and action $a_k(\xi, f) = \xi(k/f)$ for $k = 0, \ldots, K-1$. The chunk is executed at control frequency f_ξ, so action a_k corresponds to physical time $\tau_k = k/f_\xi$, with step index k serving as its positional embedding.

3.1 Continuous Action Generation via Frequency-Domain Flow Matching

Training robot foundation models requires vast amounts of data. Yet because different robot embodiments use different sensing and control stacks, trajectories are often collected at mismatched

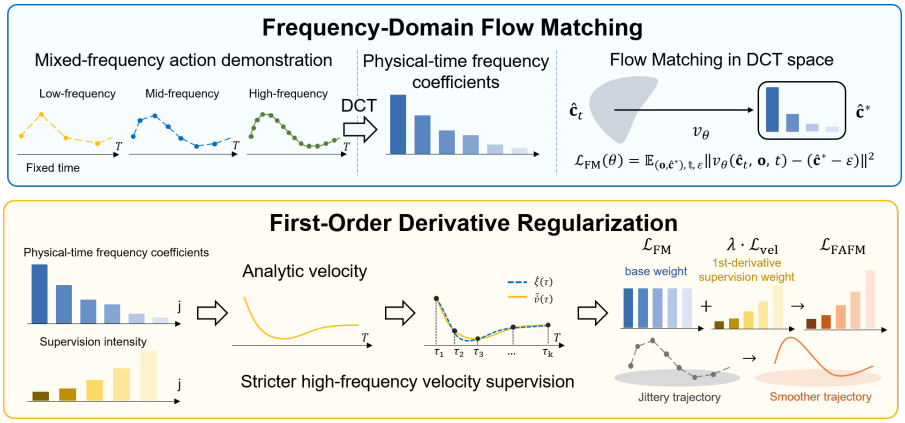

Figure 2: Overview of FAFM. Demonstration trajectories are mapped to DCT coefficients anchored to physical time, flow matching is performed in coefficient space, and analytic temporal-derivative supervision regularizes the continuous action for smooth execution.

temporal resolutions. For example, in the Open X-Embodiment dataset [10], RT-1 [11] trajectories are recorded at 3 Hz, whereas ALOHA [9] trajectories are recorded at 50 Hz. Early robot foundation models, such as π_0 [3] and $\pi_{0.5}$ [4], typically ignored these differences. Nevertheless, learning from trajectories logged at heterogeneous frequencies can be problematic: identical observation–action histories may correspond to different next-action targets once the data are indexed in time, which can induce conflicting gradient signals during optimisation. More generally, we argue that standard step-indexed action chunk learning from heterogeneous frequency training data can be ambiguous.

Proposition 1 (Gradient Conflict in Mixed-Frequency Training)**.** *Consider the step-indexed objective*

$$\mathcal{J}(\hat{a}) \;=\; \mathbb{E}_{(\mathbf{o},\xi,f)\sim p_{\mathcal{D}}}\, \mathbb{E}_{k\sim\mathrm{Unif}\{0,\dots,K(\xi,f)-1\}}\Big[\big\|\hat{a}(k \mid \mathbf{o}) - \xi(k/f)\big\|^2\Big]. \tag{2}$$

If $f \mid \mathbf{o}$ is non-degenerate, there exist $f_1 \neq f_2$ in the support of $p_{\mathcal{D}}(f \mid \mathbf{o})$, the Bayes-optimal solution

$$\hat{a}^\star(k \mid \mathbf{o}) \;=\; \mathbb{E}_{(\xi,f)\sim p_{\mathcal{D}}(\cdot|\mathbf{o},\, k<K(\xi,f))}\big[\,\xi(k/f)\,\big|\,\mathbf{o},\, k\,\big] \tag{3}$$

averages evaluations of ξ at distinct *physical times $k/f_1 \neq k/f_2$, producing a physically infeasible target that lies on no single demonstration trajectory in the support of $p_{\mathcal{D}}(\xi \mid \mathbf{o})$.*

Proof sketch: The L^2-optimal predictor is the conditional mean $\hat{a}^\star(\mathbf{o},k) = \mathbb{E}[\xi(k/f) \mid \mathbf{o}, k]$. As $\hat{a}^\star$ depends only on $p_{\mathcal{D}}$, no amount of scaling, or optimization can remove this failure. See Appendix A.

By contrast, a frequency-domain representation can accommodate trajectories sampled at different rates. We therefore replace step-indexed action chunks with DCT representations. To construct such a representation, we model the physical trajectory as an element of $\mathcal{H} := H^1([0,T_\xi];\mathbb{R}^d) := \{f \in L^2(0,T_\xi;\mathbb{R}^d) \mid f' \in L^2(0,T_\xi;\mathbb{R}^d)\}$ with inner product

$$\langle f,g\rangle_{H^1} \;=\; \langle f,g\rangle_{L^2} \;+\; \langle f',g'\rangle_{L^2}, \tag{4}$$

and expand in the L^2-orthonormal cosine basis $\phi_j(\tau) = \alpha_j \cos(\omega_j \tau)$, $\omega_j = j\pi/T_\xi$, $\alpha_0 = \sqrt{1/T_\xi}$, $\alpha_j = \sqrt{2/T_\xi}$ for $j \geq 1$, which diagonalizes $-\partial_\tau^2$ under Neumann boundary conditions. The target trajectory admits the expansion $\xi^* = \sum_{j\geq 0} c_j^* \phi_j$ with $c_j^* = \langle \xi^*, \phi_j\rangle_{L^2}$. The DCT coefficients are the L^2 projections of ξ onto this basis; their structure directly shapes the Sobolev objective in Section 3.2. To remove dependence on step index, we fix the physical chunk duration T_ξ and represent each trajectory by its DCT-II [47] coefficient vector $\hat{\mathbf{c}} \in \mathbb{R}^{(M+1)\times d}$, $M \ll K$:

$$\hat{c}_j = \frac{2}{K}\sum_{n=0}^{K-1} \xi_n \cos\Big(\frac{j\pi(2n+1)}{2K}\Big), \quad j = 0,\dots,M, \tag{5}$$

where $\xi_n = \xi^*(n/f_\xi)$ and $K = \lfloor T_\xi f_\xi \rfloor$ varies with the demonstration frequency. Because $\frac{j\pi(2n+1)}{2K} = \frac{j\pi\tau_n}{T_\xi}$ depends only on physical time $\tau_n = n/f_\xi$, Eq. (5) is a quadrature rule for the frequency-independent integral

$$c_j^* \;=\; \frac{2}{T_\xi}\int_0^{T_\xi} \xi^*(\tau)\cos\Big(\frac{j\pi\tau}{T_\xi}\Big)d\tau, \qquad \hat{c}_j = c_j^* + O(1/K). \tag{6}$$

Since c_j^* depends on ξ^* and T_ξ, instead of f_ξ, two demonstrations of the same physical motion at different control frequencies produce *identical* coefficient targets up to $O(1/K)$, directly resolving the identifiability failure in Proposition 1. Moreover, the frequency-domain coefficients serve as weights over cosine basis functions, enabling reconstruction of a continuous action trajectory executable at arbitrary temporal resolutions:

$$\hat{v}(\tau) = \tfrac{1}{2}\hat{c}_0 + \sum_{j=1}^{M} \hat{c}_j \cos(\omega_j \tau), \qquad \omega_j = \frac{j\pi}{T_\xi}, \quad \tau \in [0, T_\xi]. \tag{7}$$

At inference, the ODE is integrated in the $(M+1)\times d$-dimensional coefficient space and the action is recovered via Eq. (7). FAFM replaces the chunk-space transport target with the coefficient vector $\hat{\mathbf{c}}^*$, forming the straight-line interpolant $\hat{\mathbf{c}}_t = t\hat{\mathbf{c}}^* + (1-t)\varepsilon$, the flow-matching loss becomes

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{(\mathbf{o},\hat{\mathbf{c}}^*),\, t,\, \varepsilon} \Big\| v_\theta(\hat{\mathbf{c}}_t,\, \mathbf{o},\, t) - \big(\hat{\mathbf{c}}^* - \varepsilon\big) \Big\|^2, \tag{8}$$

which is gradient-equivalent to an ℓ^2 error on decoded position sequences (Lemma 1, Appendix B).

3.2 Temporally Consistent Actions via First-Order Derivative Regularization

During inference, adjacent actions within a chunk are executed sequentially and are therefore temporally coupled. However, most flow-matching formulations treat the actions in a chunk independently, ignoring such temporal correlations. As a result, adjacent actions may each incur only small errors but deviate in opposite directions, producing high-frequency oscillations and jitter. Such instability can be particularly costly in soft-body manipulation, causing spillage during liquid pouring and irreversible deformation during surgical tissue handling.

Beyond supervising per-timestep action values, the first-order temporal derivative of actions provides an explicit supervision signal for how actions should evolve over time. Penalizing changes between neighbouring actions yields coupled predictions within a chunk, discourages abrupt action variations, and improves temporal consistency. In discrete action chunks, this derivative is typically approximated using finite differences between adjacent actions. However, such estimates can be noisy and implicitly assume a piecewise-linear evolution between samples. In contrast, our DCT-based continuous action parameterization yields analytic temporal derivatives directly from the DCT coefficients, enabling accurate and stable derivative supervision. In particular,

$$\hat{\dot{v}}(\tau) = -\sum_{j=1}^{M} \hat{c}_j\, \omega_j \sin(\omega_j \tau), \tag{9}$$

which is exact within the M-mode subspace and free of finite-difference noise. We therefore additionally supervise the first-order derivative of actions by flow-matching:

$$\mathcal{L}_{\mathrm{vel}}(\theta) = \mathbb{E}_{(\xi,\tau_k)} \big\| \hat{\dot{v}}(\tau_k) - \dot{\xi}(\tau_k) \big\|^2, \tag{10}$$

where $\dot{\xi}(\tau_k)$ is the first-order derivative of the demonstration. The total objective is

$$\mathcal{L}_{FAFM} = \mathcal{L}_{\mathrm{FM}} + \lambda \cdot \mathcal{L}_{\mathrm{vel}}, \qquad \lambda = 1. \tag{11}$$

Empirically, we find that $\lambda = 1$ already yields good result across tasks and use this value throughout our paper. Accordingly, FAFM introduces no additional networks and can be applied off-the-shelf.

Understanding the loss. We further find our FAFM loss corresponds to squared H^1_μ Sobolev norm, and thus enhancing smoothness by penalizing high-frequency error, empirically speeds up convergence and increases robustness against mechanical bias. Specifically, let $\delta_j := \hat{c}_{\theta,j} - c_j^*$ denote coefficient prediction error. Then Lemma 1 (Appendix B) gives $\mathcal{L}_{\mathrm{FM}} = \sum_{j=0}^{M} \delta_j^2$, while DST orthogonality gives $\mathcal{L}_{\mathrm{vel}} = \sum_{j=1}^{M} \omega_j^2\, \delta_j^2$. Hence,

$$\mathcal{L}_{FAFM}(\theta) \;=\; \sum_{j=0}^{M} \mu_j\, \delta_j^2, \qquad \mu_j \;:=\; 1 + \lambda \omega_j^2, \tag{12}$$

which is the squared H^1_μ Sobolev norm on the projection residual defined below $\hat{\xi}_\theta - P_M \xi^*$. Hence,

Theorem 1 (FAFM loss as a weighted H^1-projection error)**.** *Let $V_M := \mathrm{span}\{\phi_0, \ldots, \phi_M\} \subset H^1([0, T_\xi]; \mathbb{R}^d)$, let $P_M : L^2 \to V_M$ denote L^2-orthogonal projection, and let $\hat{\xi}_\theta = \sum_{j=0}^{M} \hat{c}_{\theta,j}\phi_j \in V_M$ be the FAFM-decoded prediction. Under centered-grid sampling $\tau_n = (n + \frac{1}{2})/f_\xi$ and finite-difference velocity targets,*

$$\mathcal{L}_{FAFM}(\theta) \;=\; \left\|\hat{\xi}_\theta - P_M\xi^*\right\|_{H^1_\mu}^2, \qquad \|f\|_{H^1_\mu}^2 \;:=\; \sum_{j\geq 0}(1 + \lambda\omega_j^2)\left|\langle f, \phi_j\rangle_{L^2}\right|^2. \tag{13}$$

Proof sketch: Centered-grid DCT orthogonality gives $\mathcal{L}_{\mathrm{FM}} = \sum_j \delta_j^2$, while DST orthogonality gives $\mathcal{L}_{\mathrm{vel}} = \sum_j \omega_j^2\delta_j^2$. Eq. (12) *then follows immediately; the full proof is in Appendix B.*

Remark 1: Higher-order dynamics correction. The ω_j^2 weight in μ_j supplies stronger gradient pressure on high-frequency coefficient errors than standard L^2 flow matching, directly suppressing high-frequency artifacts such as action jitter. A formal mode-wise efficiency analysis across derivative orders is given in Appendix C.

Remark 2: Convergence acceleration. For physically smooth trajectories, μ_j acts as a built-in spectral preconditioner: it encourage higher gradient on high-frequency component, which empirically accelerate convergence. A formal analysis is available in Appendix E.

Remark 3: Robustness to embodiment mechanical bias. A constant mechanical offset perturbs only the DC coefficient under FAFM's representation, leaving all shape-defining coefficients exactly unchanged. In contrast, a constant offset perturb all actions in flow matching chunks, causing greater harm to the robustness of flow matching policies. See Appendix F for a derivation.

4 Experiments

Evaluation environments: We evaluate FAFM in two stages. First, as a standalone policy, it is assessed on synthetic toy benchmark, obstacle avoidance, LapGym [18], and deployed on a real Franka robot. Second, FAFM is integrated into VLA backbones, evaluated on LIBERO [19], and again deployed on a real Franka robot. The specific purpose of each environment is described below.

Evaluation metrics: We adopt success rate (SR) as the primary measure of task reliability. Motion smoothness is assessed using the log dimensionless jerk (LDLJ), computed over successful episodes, which is widely used in robot-assisted surgery and surgical skill assessment [48, 49, 50]. LDLJ is defined on a logarithmic scale, so small numerical differences reflect exponential improvement in the underlying motion smoothness. For both metrics, higher values indicate better performance.

Baselines: In the standalone-policy setting, our baselines include Diffusion Policy (DP) [20], Flow Matching Policy (FM) [27], Streaming Flow Policy (SFP) [46], FreqPolicy [45], and MPD [12], with our method implemented on flow matching backbone. In the VLA setting, we use π_0 [3] or $\pi_{0.5}$ [4] as the backbone, our method is implemented by changing their action head to our FAFM-based variant. See implementation details and additional training details in Appendix G.1.

4.1 Standalone Policy Experiments

4.1.1 Synthetic Toy Benchmark Results

We first verify the ability to learn smooth, multimodal actions on a simple 1D task. The demonstrations comprise two crossing sinusoidal modes, as shown in Fig. 3. While our FAFM solves this task easily, existing methods struggle on this simple setting. FM preserves both modes but yields jittery trajectories. SFP fails to separate the two modes because its policy conditions on only a single previous timestep, which is insufficient when the modes overlap. FreqPolicy exhibits higher bias and occasionally produces trajectory with lower amplitude, as its low-frequency anchor is entangled with observation features in latent space. MPD captures global trajectory structure but over-smooths local modes, resulting in lower amplitude output. In contrast, FAFM cleanly separates the two sinusoidal modes while producing smooth trajectories consistently.

4.1.2 Obstacle Avoidance Results

Settings. Built on PyBullet, the obstacle-avoidance task requires a robot arm to navigate across multiple obstacles. Because many valid paths exist, it is a standard benchmark for evaluating solution

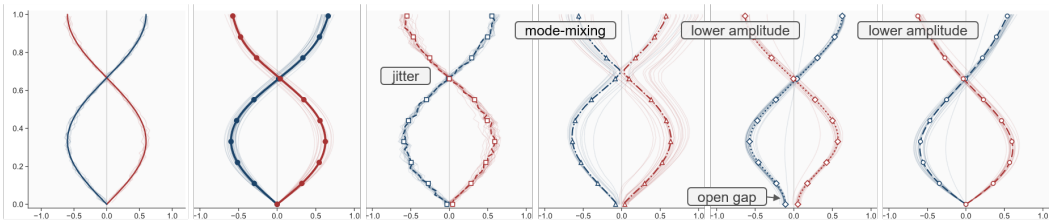

Figure 3: Synthetic toy bi-modal trajectory experiment. FAFM uniquely capture the multimodal modes and yields smooth trajectory.

Method	SR	LDLJ	M
DP	35	-9.16 ± 0.77	8
SFP	49	-6.98 ± 0.82	3
MPD	16	-6.78 ± 0.47	2
FreqPolicy	39	-9.02 ± 1.11	10
FM	48	-8.62 ± 0.69	14
ours	**61**	**-5.60±1.08**	12

Table 1: Avoidance results.

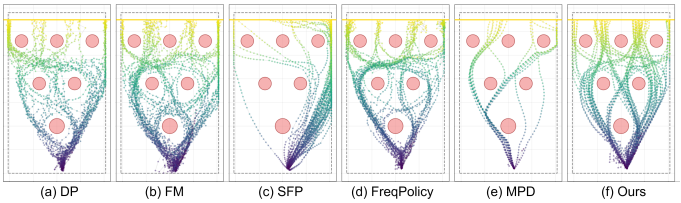

Figure 4: Successful obstacle avoidance trajectories.

diversity. In addition to success rate and smoothness, we report the number of distinct trajectory modes (M), defined as modes traversed by more than two successful trajectories.

Results. As shown in Table 1, FAFM simultaneously achieves high success rate, motion smoothness, and solution diversity. SFP and MPD produce smooth trajectories but fail to learn multimodal policies, resulting in limited path diversity. Conversely, DP, FreqPolicy, and FM discover diverse solutions but generate jittery actions, as reflected by lower LDLJ. These observations are corroborated qualitatively in Fig. 4: the solutions of SFP and MPD are confined to a narrow set of paths, whereas those of DP, FM, and FreqPolicy are noisy and unstable. In contrast, FAFM is the only method achieving both smoothness and diversity.

4.1.3 LapGym: Surgical Manipulation Results

Settings. LapGym is an open-source simulation environment for robot-assisted laparoscopic surgery. In LapGym, the robot operate on deformable objects, such as tissues and suture threads, demanding gentle and precise manipulation. As shown in Fig. 5(a), we evaluate on four tasks from this environment: rope threading (RT), grasp-lift-touch (GLT), bimanual tissue manipulation (BTM), and ligating loop (LL). All methods are trained for 3,000 epochs for their best performance.

Results. As shown in Table 2, FAFM achieves the highest success rate and motion smoothness across all four tasks. Although MPD also yields strong results, its smoothness relies on hand-specified ProDMP priors, whereas FAFM requires no such prior. Furthermore, as shown in Fig. 5(b), FAFM converges faster than all baselines, which is a practically important advantage in time-constrained settings such as real-world RL.

4.1.4 Ablation Study

We further evaluate the contribution of each component of FAFM on the rope-threading task as shown in Fig. 6. Removing the DCT transform (ours w/o DCT) reduce success rates and LDLJ to flow-matching backbone, indicating that frequency-domain modelling is the dominant contributor to overall performance. Notably, ours w/o DCT also shows negligible improvement in LDLJ, suggesting that supervising the first-order derivative via finite differences is ineffective without the continuous frequency-domain parameterization. Removing the derivative supervision (ours w/o $\mathcal{L}_{\text{vel}}$) results in a moderate drop in both success rate and LDLJ, confirming the benefit of supervising the first-order derivative as H^1_μ Sobolev regularization.

4.1.5 Real-Robot Evaluation

Finally, we deploy all methods on a Franka physical robot arm for pick-and-place, which requires picking a cup and place it on the plate. For all methods, we fine-tune it on 40 demos and evaluate the result over 20 episodes. As shown in Figure 7, While our FAFM and MPD both achieves 100% success rate, out FAFM achieve better motion smoothness measured by LDLJ. It is also worth noting

Table 2: LapGym results. LDLJ is reported as mean ± std. **Bold**: best per column.

	RT		GLT		BTM		LL	
Method	SR	LDLJ	SR	LDLJ	SR	LDLJ	SR	LDLJ
DP	92	-10.88 ± 1.61	**100**	-16.62 ± 0.52	96	-6.05 ± 2.18	**100**	-13.04 ± 0.93
SFP	72	-11.54 ± 2.26	13	-17.09 ± 1.44	95	-5.62 ± 2.53	99	-12.46 ± 1.33
MPD	94	-9.38 ± 2.12	99	-14.37 ± 0.52	**100**	-3.81 ± 1.92	**100**	-11.47 ± 1.00
FreqPolicy	89	-9.33 ± 1.40	**100**	-15.52 ± 0.55	83	-4.71 ± 1.88	**100**	-12.49 ± 0.90
FM	89	-10.69 ± 2.06	**100**	-16.62 ± 0.60	94	-5.82 ± 1.87	**100**	-13.23 ± 0.87
FAFM	**97**	$\mathbf{-7.57 \pm 1.32}$	**100**	$\mathbf{-14.31 \pm 0.58}$	99	$\mathbf{-2.21 \pm 2.34}$	**100**	$\mathbf{-11.28 \pm 1.17}$

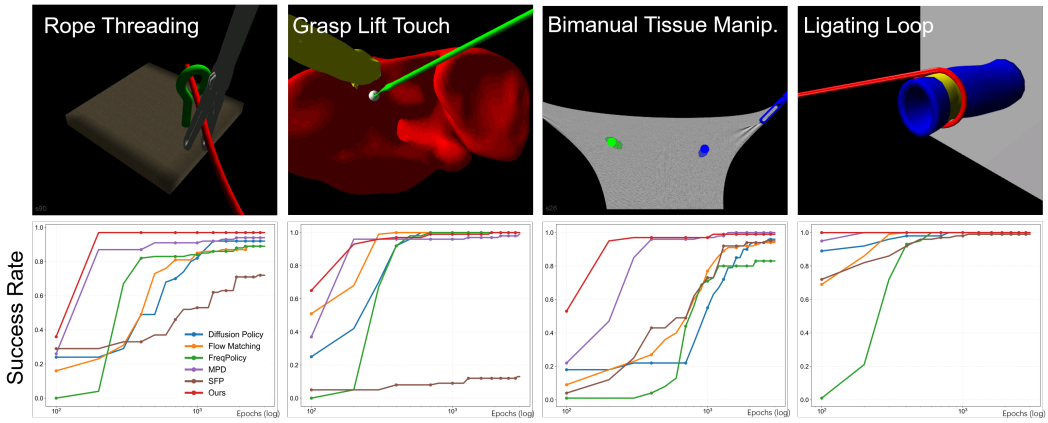

Figure 5: **Top row**: Overview of surgical manipulation tasks in LapGym. **Bottom row**: Success rate on LapGym tasks as training proceeds. Our FAFM converge to the best result earlier than baselines.

that SFP drops to 50% SR on the physical system, likely due to its marginal-only trajectory modeling being sensitive to real-world observation noise.

4.2 VLA Scaling Experiments

4.2.1 LIBERO Results

FAFM enhance smoothness and is robust against mechanical bias. We evaluate the performance of FAFM on LIBERO, a commonly used benchmark for VLAs, implementing our method on π_0 and $\pi_{0.5}$ backbone and treating the unmodified VLAs as baselines. As shown in Figure 8, FAFM achieves comparable success rates on both backbones while consistently improving motion smoothness. Under mechanical bias, FAFM yields substantially higher success rates alongside improved smoothness on both backbones, confirming that our DCT-based action parameterization is robust to constant action bias.

FAFM supports mixed-frequency input. Training VLAs typically requires large datasets collected across diverse hardware platforms, which are often recorded at heterogeneous control frequencies. To evaluate robustness to this setting, we conduct a controlled experiment on the LIBERO drawer task and real-robot pick-and-place task with 43 mixed-frequency demonstrations at 5/10/20 Hz.FAFM fixes $T_\xi = 5\,\mathrm{s}$, mapping $K = 25/50/100$ samples to $M + 1 = 17$ DCT coefficients. We compare against π_0-Resample, which linearly resamples demonstrations to $10\,\mathrm{Hz}$, and π_0-FreqCond, which conditions on recording frequency as in RDT-1B [36]. As shown in Fig. 9, FAFM achieves higher success rates than both baselines in simulation and real-world experiments, with improved motion smoothness reported in Appendix G.2.

4.3 Real-Robot Deployment with $\pi_{0.5}$

Finally, we evaluate the performance of our method on $\pi_{0.5}$ backbone. We include two tasks for comparison. The first one is pick-and-place with variable object positions and goal configurations, and the second is multi-obstacle avoidance with slightly perturbed obstacle positions and start points, see Fig. 10 for illustrations. We collect 20 demos for pick-and-place and 60 demos for multi-obstacle avoidance, which is sufficient for their convergence, and evaluate each method and each task for 20 episodes. Our method achieves $+15\%$ success rate on pick-and-place and $+10\%$ success rate on

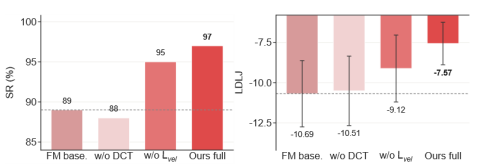

Figure 6: Ablation on rope threading.

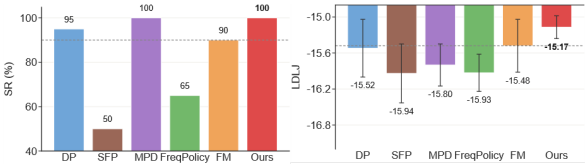

Figure 7: Real-robot policy pick&place.

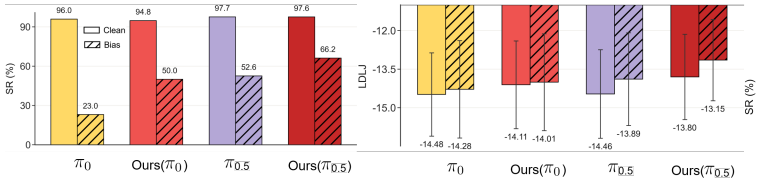

Figure 8: LIBERO results. "Clean" is the unperturbed baseline. Bias denotes mechanical offset.

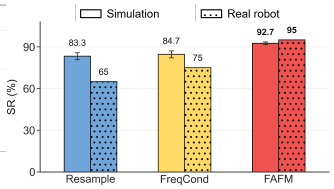

Figure 9: mixed-freq results.

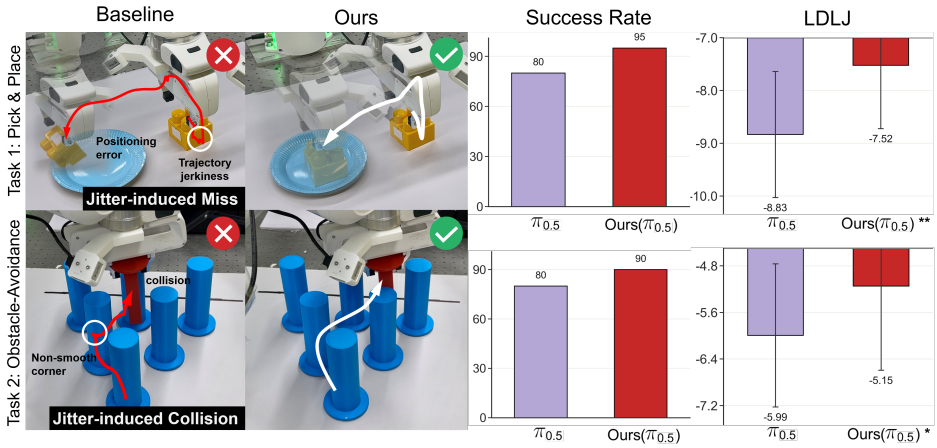

Figure 10: Real-world deployment with the $\pi_{0.5}$ backbone. Our FAFM achieves higher success rate by improved motion smoothness (LDLJ), which avoids jitter-induced miss and collisions. For LDLJ, paired samples t-test shows $p < 0.001(^{**})$ for Task 1, $p < 0.05$ $(^{*})$ for Task 2.

obstacle avoidance comparing with $\pi_{0.5}$ backbone. The higher success rate comes from improved motion smoothness, which better avoids jitter-induced target missing and collisions in both tasks.

5 Conclusion

We presented FAFM, a frequency-aware flow-matching framework that enables continuous and temporally consistent action generation for robotic manipulation. By performing flow matching in frequency space via DCT and regularizing the first-order derivative, our method supports mixed-frequency training and provides smooth actions. Our FAFM is simple, free of additional learnable parameters, and applies to standalone flow matching policy and VLAs. Experimental results across synthetic toy benchmarks, obstacle avoidance, LapGym, and LIBERO demonstrate consistent improvements over strong baselines in success rates, multimodal expressivity, motion smoothness, convergence speed, robustness to mechanical bias and mixed-frequency input. These advantages are further validated on real-world Franka robot. As for **limitations**, our approach is less effective for tasks that intrinsically require high-frequency or impulsive actions due to the nature of DCT, such as sharp jumps or percussive interactions. However, the vast majority of real-world robotic manipulation tasks are characterized by smooth, low-frequency action trajectories suitable for our FAFM.

Acknowledgments and Disclosure of Funding

This work described in this paper was supported in part by a grant from the NSFC/RGC Joint Research Scheme sponsored by the Research Grants Council of the Hong Kong Special Administrative Region, China and the National Natural Science Foundation of China (Project No. N_CUHK410/23) and by National Natural Science Foundation of China (Project No. 62606026, 62322318). This work was also supported by Beijing Natural Science Foundation (Project No. QY26142).

References

[1] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*.

[2] Qiang Liu. Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint arXiv:2209.14577*, 2022.

[3] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *eprint arXiv: 2410.24164*, 2024.

[4] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_0.5$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

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

[6] Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025.

[7] Kaidong Zhang, Jian Zhang, Rongtao Xu, Yu Sun, Shuoshuo Xue, Youpeng Wen, Xiaoyu Guo, Minghao Guo, Weijia Liufu, Liu Zihou, et al. A1: A fully transparent open-source, adaptive and efficient truncated vision-language-action model. *arXiv preprint arXiv:2604.05672*, 2026.

[8] Chilam Cheang, Sijin Chen, Zhongren Cui, Yingdong Hu, Liqun Huang, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Xiao Ma, et al. Gr-3 technical report. *arXiv preprint arXiv:2507.15493*, 2025.

[9] 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.

[10] 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.

[11] 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.

[12] Paul Maria Scheikl, Nicolas Schreiber, Christoph Haas, Niklas Freymuth, Gerhard Neumann, Rudolf Lioutikov, and Franziska Mathis-Ullrich. Movement primitive diffusion: Learning gentle robotic manipulation of deformable objects. *IEEE Robotics and Automation Letters*, 9(6):5338–5345, 2024.

[13] Khang Nguyen, An T Le, Tien Pham, Manfred Huber, Jan Peters, and Minh Nhat Vu. Flowmp: Learning motion fields for robot planning with conditional flow matching. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11291–11297. IEEE, 2025.

[14] Fan Yang, Peiguang Jing, Kaihua Qu, Ningyuan Zhao, and Yuting Su. Abpolicy: Asynchronous b-spline flow policy for real-time and smooth robotic manipulation. *arXiv preprint arXiv:2602.23901*, 2026.

[15] Aleks Attanasio, Bruno Scaglioni, Elena De Momi, Paolo Fiorini, and Pietro Valdastri. Autonomy in surgical robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 4(1):651–679, 2021.

[16] Nasir Ahmed, T_ Natarajan, and Kamisetty R Rao. Discrete cosine transform. *IEEE transactions on Computers*, 100(1):90–93, 1974.

[17] 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. *arXiv preprint arXiv:2501.09747*, 2025.

[18] Paul Maria Scheikl, BalĂĄzs Gyenes, Rayan Younis, Christoph Haas, Gerhard Neumann, Martin Wagner, and Franziska Mathis-Ullrich. Lapgym-an open source framework for reinforcement learning in robot-assisted laparoscopic surgery. *Journal of Machine Learning Research*, 24(368):1–42, 2023.

[19] 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.

[20] 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.

[21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[22] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[23] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. *arXiv preprint arXiv:2403.03954*, 2024.

[24] Julen Urain, Niklas Funk, Jan Peters, and Georgia Chalvatzaki. Se (3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion. In *2023 IEEE international conference on robotics and automation (ICRA)*, pages 5923–5930. IEEE, 2023.

[25] Moritz Reuss, Maximilian Li, Xiaogang Jia, and Rudolf Lioutikov. Goal-conditioned imitation learning using score-based diffusion policies. *arXiv preprint arXiv:2304.02532*, 2023.

[26] Dian Wang, Stephen Hart, David Surovik, Tarik Kelestemur, Haojie Huang, Haibo Zhao, Mark Yeatman, Jiuguang Wang, Robin Walters, and Robert Platt. Equivariant diffusion policy. *arXiv preprint arXiv:2407.01812*, 2024.

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

[28] Qinglun Zhang, Zhen Liu, Haoqiang Fan, Guanghui Liu, Bing Zeng, and Shuaicheng Liu. Flowpolicy: Enabling fast and robust 3d flow-based policy via consistency flow matching for robot manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 14754–14762, 2025.

[29] Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface. *arXiv preprint arXiv:2407.15208*, 2024.

[30] Max Braun, Noémie Jaquier, Leonel Rozo, and Tamim Asfour. Riemannian flow matching policy for robot motion learning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5144–5151. IEEE, 2024.

[31] Minho Park, Kinam Kim, Junha Hyung, Hyojin Jang, Hoiyeong Jin, Jooyeol Yun, Hojoon Lee, and Jaegul Choo. Acg: Action coherence guidance for flow-based vla models. *arXiv preprint arXiv:2510.22201*, 2025.

[32] 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.

[33] 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.

[34] 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.

[35] Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024.

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

[37] Yifan Zhong, Xuchuan Huang, Ruochong Li, Ceyao Zhang, Zhang Chen, Tianrui Guan, Fanlian Zeng, Ka Nam Lui, Yuyao Ye, Yitao Liang, et al. Dexgraspvla: A vision-language-action framework towards general dexterous grasping. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 18836–18844, 2026.

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

[39] Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, et al. $\pi^*_{0.6}$: a vla that learns from experience. *arXiv preprint arXiv:2511.14759*, 2025.

[40] Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, et al. $\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026.

[41] Haoyun Liu, Jianzhuang Zhao, Xinyuan Chang, Tianle Shi, Chuanzhang Meng, Jiayuan Tan, Feng Xiong, Tong Lin, Dongjie Huo, Mu Xu, et al. Neural implicit action fields: From discrete waypoints to continuous functions for vision-language-action models. *arXiv preprint arXiv:2603.01766*, 2026.

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

[43] Xirui Shi and Jun Jin. Frmd: Fast robot motion diffusion with consistency-distilled movement primitives for smooth action generation. *arXiv preprint arXiv:2503.02048*, 2025.

[44] Ge Li, Zeqi Jin, Michael Volpp, Fabian Otto, Rudolf Lioutikov, and Gerhard Neumann. Prodmp: A unified perspective on dynamic and probabilistic movement primitives. *IEEE Robotics and Automation Letters*, 8(4):2325–2332, 2023.

[45] Yiming Zhong, Yumeng Liu, Chuyang Xiao, Zemin Yang, Youzhuo Wang, Yufei Zhu, Ye Shi, Yujing Sun, Xinge Zhu, and Yuexin Ma. Freqpolicy: Frequency autoregressive visuomotor policy with continuous tokens. *arXiv preprint arXiv:2506.01583*, 2025.

[46] Sunshine Jiang, Xiaolin Fang, Nicholas Roy, Tomás Lozano-Pérez, Leslie Pack Kaelbling, and Siddharth Ancha. Streaming flow policy: Simplifying diffusion/flow-matching policies by treating action trajectories as flow trajectories. *arXiv preprint arXiv:2505.21851*, 2025.

[47] Gilbert Strang. The discrete cosine transform. *SIAM review*, 41(1):135–147, 1999.

[48] Simar Singh, Joe Bible, Zhanhe Liu, Ziyang Zhang, and Ravikiran Singapogu. Motion smoothness metrics for cannulation skill assessment: What factors matter? *Frontiers in Robotics and AI*, 8:625003, 2021.

[49] Farzad Aghazadeh and Bin Zheng. A new approach to laparoscopic skill assessment: Motion smoothness and bimanual coordination. *Laparoscopic, Endoscopic and Robotic Surgery*, 8(2):90–95, 2025.

[50] Ariel Rodriguez, Lorenzo Mazza, Martin Lelis, Rayan Younis, Sebastian Bodenstedt, Martin Wagner, and Stefanie Speidel. An open-source robotics research platform for autonomous laparoscopic surgery. *arXiv preprint arXiv:2603.08490*, 2026.

[51] Shuhan Xue, Zixin Ding, Yichen Shen, Yinjie Wang, Zhenfei Yin, Yingcheng Wu, Yuxin Chen, Mengdi Wang, and Ling Yang. Past-bench: Benchmarking the foundations of recursive self-improvement in personal agents. *arXiv preprint arXiv:2608.04003*, 2026.

[52] Kevin Black, Manuel Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. *Advances in Neural Information Processing Systems*, 38:33383–33407, 2026.

A Proof of Proposition 1

Proof. Fix $(\mathbf{o}, k)$ and write $Y = \xi(k/f) \cdot \mathbf{1}\{k < K(\xi, f)\}$ for the regression target restricted to the event that token k exists. The conditional expectation $\mathbb{E}[Y \mid \mathbf{o}, k]$ is the L^2-projection of Y onto the σ-algebra generated by $(\mathbf{o}, k)$, so for any measurable predictor $\hat{a}(\mathbf{o}, k)$,

$$\mathbb{E}\big[(\hat{a} - Y)^2 \mid \mathbf{o}, k\big] \;=\; \big(\hat{a}(\mathbf{o}, k) - \mathbb{E}[Y \mid \mathbf{o}, k]\big)^2 \;+\; \mathrm{Var}(Y \mid \mathbf{o}, k),$$

in which only the first term depends on $\hat{a}$. Minimization gives $\hat{a}^\star(\mathbf{o}, k) = \mathbb{E}[Y \mid \mathbf{o}, k]$, which is (3).

Under the non-degeneracy assumption, the integrand $\xi(k/f)$ in (3) takes values $\xi(k/f_1), \xi(k/f_2), \ldots$ at *different* physical times $k/f_1 \neq k/f_2$ across demonstrations. The expectation therefore averages evaluations of ξ at *distinct phases of the underlying motion*, not at the same phase of different motions. A convex combination of points sampled from distinct phases of even a single trajectory ξ generally falls off the trajectory itself; a fortiori the same holds when the points are drawn from different trajectories at different phases.

□

Remark 1. *The predictor $\hat{a}(k \mid \mathbf{o})$ above depends on $(\mathbf{o}, k)$ but don't have the information about f.*

Remark 2. *The function $\hat{a}^\star$ in* (3) *depends only on the conditioning $(\mathbf{o}, k)$ and the distribution $p_{\mathcal{D}}$; it is independent of training-set size, model capacity, and optimization procedure.Because the right-hand side of* (3) *is a deterministic functional of $p_{\mathcal{D}}$ alone. Increasing the training set drives the empirical risk to its population infimum $\mathcal{J}(\hat{a}^\star)$; increasing model capacity makes the empirical minimizer closer to $\hat{a}^\star$ in L^2; neither changes $\hat{a}^\star$ itself. The optimization procedure determines how closely the trained model approximates $\hat{a}^\star$, but cannot reach a target lower than $\mathcal{J}(\hat{a}^\star)$.*

Extension to Flow Matching. For straight-interpolation conditional flow matching, let $A_t = tA + (1-t)\varepsilon$, $Z = (A_t, o, t)$, and $U = A - \varepsilon$. The FM objective is

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}\left[\|v_\theta(Z) - U\|^2\right], \tag{14}$$

whose optimum is $v^*(Z) = \mathbb{E}[U \mid Z]$. When the recording frequency F is unobserved, the law of total variance gives

$$\mathrm{Var}(U \mid Z) = \mathbb{E}[\mathrm{Var}(U \mid Z, F) \mid Z] + \mathrm{Var}(\mathbb{E}[U \mid Z, F] \mid Z). \tag{15}$$

The second term captures additional between-frequency target variance. Thus, frequency-agnostic step-indexed flow matching can introduce conflicting gradient signals when targets from different recording frequencies overlap.

B Proof of Theorem 1

Before proving Theorem 1, we establish that the position-space implementation of $\mathcal{L}_{\mathrm{FM}}$ used in practice is gradient-equivalent to the coefficient-space form in Eq. (8).

Lemma 1 (Implementation equivalence)**.** *Let $\hat{\mathbf{c}}_\theta \in \mathbb{R}^{M+1}$ be the coefficient-space prediction and $\hat{\xi}_\theta(\tau_n) = \sum_{j=0}^{M} \hat{c}_{\theta,j}\phi_j(\tau_n)$ its decoded trajectory at the centered grid $\tau_n = (n + \frac{1}{2})/f_\xi$. Let $\hat{\mathbf{c}}^* \in \mathbb{R}^{M+1}$ be the target coefficients and $\xi_n^* := \xi^*(\tau_n)$ the target chunk. Then*

$$\sum_{n=0}^{K-1} \big(\hat{\xi}_\theta(\tau_n) - \xi_n^*\big)^2 \;=\; \frac{K}{T_\xi} \sum_{j=0}^{M} \big(\hat{c}_{\theta,j} - c_j^*\big)^2 \;+\; R_M(\xi^*), \tag{16}$$

where $R_M(\xi^*) = \sum_{n=0}^{K-1} \big(\xi_n^* - (P_M\xi^*)(\tau_n)\big)^2$ depends only on the high-frequency tail $\{c_j^*\}_{j>M}$ of the target trajectory and not on θ.

Proof. Decompose the chunk-space residual at each sample point as

$$\hat{\xi}_\theta(\tau_n) - \xi_n^* \;=\; \underbrace{\hat{\xi}_\theta(\tau_n) - (P_M\xi^*)(\tau_n)}_{\in V_M} \;+\; \underbrace{(P_M\xi^*)(\tau_n) - \xi_n^*}_{\perp V_M \text{ in } L^2}, \tag{17}$$

where the first term is in V_M and equals $\sum_{j=0}^{M} \delta_j \phi_j(\tau_n)$, and the second is the truncation tail $-\sum_{j>M} c_j^* \phi_j(\tau_n)$. On the centered grid, the discrete inner product $\langle f, g \rangle_K := \sum_{n=0}^{K-1} f(\tau_n) g(\tau_n)$ satisfies the discrete orthogonality $\langle \phi_i, \phi_j \rangle_K = (K/T_\xi)\boldsymbol{\delta_{ij}}$ for $0 \le i, j \le K-1$ ($\boldsymbol{\delta_{ij}}$ is the Kronecker symbol), so squaring (17) and summing over n gives

$$\sum_{n=0}^{K-1} \big(\hat{\xi}_\theta(\tau_n) - \xi_n^*\big)^2 \;=\; \frac{K}{T_\xi} \sum_{j=0}^{M} \delta_j^2 \;+\; R_M(\xi^*), \tag{18}$$

where the cross term vanishes by orthogonality between modes $j \le M$ (in-V_M part) and modes $j > M$ (truncation tail), and $R_M(\xi^*) = \sum_{n=0}^{K-1} \big(\xi_n^* - (P_M \xi^*)(\tau_n)\big)^2$ is θ-independent. □

From Lemma 1, it can be noticed that the left side of equation 16 is the chunk space training loss $\mathcal{L}^{\text{chunk}}(\theta)$, and the coefficient-space training loss can be written as $\mathcal{L}^{\text{coef}}(\theta) := \sum_{j=0}^{M} \delta_j^2$. So equation 16 can also be written as:

$$\mathcal{L}^{\text{chunk}}(\theta) \;=\; \frac{K}{T_\xi} \sum_{j=0}^{M} \delta_j^2 \;+\; R_M(\xi^*) \;=\; \frac{K}{T_\xi}\, \mathcal{L}^{\text{coef}}(\theta) \;+\; R_M(\xi^*), \tag{19}$$

Consequently, since $R_M(\xi^*)$ is θ-independent and $K/T_\xi > 0$, the two losses share the same minimizer and the same gradient direction in $\hat{\mathbf{c}}_\theta$.

Here we formally start to proof Theorem 1.

Proof. From Lemma 1 we have Equ.19, with $R_M(\xi^*)$ independ on θ. We adopt the convention that the flow-matching loss is the coefficient-space form $\mathcal{L}_{\text{FM}}(\theta) := \mathcal{L}^{\text{coef}}(\theta)$, which is equivalent to the chunk-space form up to the θ-independent affine transformation in Equ.19. Thus

$$\mathcal{L}_{\text{FM}}(\theta) \;=\; \sum_{j=0}^{M} \delta_j^2, \tag{20}$$

and the same θ minimizes both $\mathcal{L}_{\text{FM}}$ and $\mathcal{L}^{\text{chunk}}$.

The model velocity is the analytic derivative $\dot{\hat{\xi}}_\theta(\tau) = -\sum_{j=1}^{M} \hat{c}_{\theta,j} \alpha_j \omega_j \sin(\omega_j \tau)$. Under the finite-difference target convention, the demonstration velocity admits the matching expansion $\dot{\xi}^*(\tau_n) = -\sum_{j=1}^{M} c_j^* \alpha_j \omega_j \sin(\omega_j \tau_n) + r_n$, where r_n collects the truncation tail $j > M$ and the finite-difference discretization error. r_n is $\langle \cdot, \cdot \rangle_K$-orthogonal to $\{\sin(\omega_j \tau_n)\}_{j=1}^{M}$ up to $O(1/K^2)$ and contributes only a θ-independent additive constant to the loss, which we absorb into a remainder $R_M^{\text{vel}}(\xi^*)$ analogous to $R_M(\xi^*)$ in Lemma 1.

The velocity-domain MSE then expands as

$$\begin{aligned}
\sum_{n=0}^{K-1} \big(\dot{\hat{\xi}}_\theta(\tau_n) - \dot{\xi}^*(\tau_n)\big)^2 &= \sum_{n=0}^{K-1} \Big(\sum_{j=1}^{M} \delta_j \alpha_j \omega_j \sin(\omega_j \tau_n)\Big)^2 \;+\; R_M^{\text{vel}}(\xi^*) \\
&= \sum_{i,j=1}^{M} \delta_i \delta_j \, \alpha_i \alpha_j \, \omega_i \omega_j \sum_{n=0}^{K-1} \sin(\omega_i \tau_n) \sin(\omega_j \tau_n) \;+\; R_M^{\text{vel}} \\
&= \frac{K}{T_\xi} \sum_{j=1}^{M} \omega_j^2 \, \delta_j^2 \;+\; R_M^{\text{vel}},
\end{aligned} \tag{21}$$

Note that here by DST orthogonality, $\sum_n sin(\omega_i \cdot \tau_n) sin(\omega_j \cdot \tau_n) = (K/2)\boldsymbol{\delta_{ij}}$, $\alpha_j^2 = 2/T_\xi$, which gives $\alpha_j^2 \cdot K/2 = K/T_\xi$ for each diagonal term. We define the coefficient-space velocity loss

$$\mathcal{L}_{\text{vel}}(\theta) \;:=\; \frac{T_\xi}{K} \sum_{n=0}^{K-1} \big(\dot{\hat{\xi}}_\theta(\tau_n) - \dot{\xi}^*(\tau_n)\big)^2 \;-\; \frac{T_\xi}{K} R_M^{\text{vel}} \;=\; \sum_{j=1}^{M} \omega_j^2 \, \delta_j^2, \tag{22}$$

which is equivalent to the implementation-level velocity MSE up to a global rescaling and a θ-independent constant.

So we have

$$\mathcal{L}_{FAFM}(\theta) \;=\; \mathcal{L}_{\mathrm{FM}}(\theta) + \lambda\,\mathcal{L}_{\mathrm{vel}}(\theta) \;=\; \sum_{j=0}^{M} \big(1 + \lambda\,\omega_j^2\big)\,\delta_j^2. \tag{23}$$

□

Finite-Sampling Approximation. We evaluate the approximation gap on 1715 Rope Threading chunks. The implemented finite-difference objective is 0.03847, compared with 0.03714 for the ideal spectral objective, corresponding to a 3.59% deviation. The coefficient-gradient cosine similarity is 0.919 ± 0.060, and the mean gradient norms differ by only 0.8%.

We further compare the same physical trajectories sampled at 10 Hz ($K = 12$) and 5 Hz ($K = 6$), keeping $T_\xi = 1.2\,\mathrm{s}$ and $M = 4$ fixed. The coefficient-target RMSE is 0.02685, and the reconstructed trajectory RMSE is 0.00947 (2.51% relative).

C Higher-Order Dynamics Correction Efficiency

Given a per-mode loss weight w_j, the gradient pressure on mode j is $2w_j\delta_j$. The *correction efficiency at order s* is the ratio between gradient pressure and the resulting s-th-derivative error amplitude:

$$\eta_j^{(s)} \;:=\; \frac{w_j}{A_j^{(s)}} \;=\; \frac{w_j}{\omega_j^s}. \tag{24}$$

A larger $\eta_j^{(s)}$ means stronger gradient pressure per unit of s-th-derivative error—the loss is more effective at correcting errors of dynamic order s at frequency ω_j. Comparing standard chunk-space FM ($w_j^{\mathrm{FM}} = 1$) with FAFM ($w_j^{FAFM} = \mu_j = 1 + \lambda\omega_j^2$):

	Position (s=0)	Velocity (s=1)	Acceleration (s=2)	Jerk (s=3)
Amplification $A_j^{(s)}$	1	ω_j	ω_j^2	ω_j^3
$\eta_j^{(s),\,\mathrm{FM}}$	1	$\propto 1/j$	$\propto 1/j^2$	$\propto 1/j^3$
$\eta_j^{(s),\,FAFM}$	$1 + \lambda\omega_j^2$	$\propto j$	$\to \lambda$	$\propto 1/j$
High-freq. ratio	$O(j^2)$	$O(j^2)$	$O(j^2)$	$O(j^2)$

Standard FM exhibits $\eta_j^{(s)} \propto 1/j^s$, decaying at every order, while FAFM replaces this decay with the high-frequency behaviors shown above. The result is a uniform $O(j^2)$ improvement over standard FM at every derivative order, with two qualitative consequences:

Velocity (s=1): FAFM gives *growing* correction efficiency $\propto j$, while standard FM *decays* as $1/j$. High-frequency velocity errors are corrected aggressively under FAFM and largely ignored under standard FM.

Acceleration (s=2): FAFM's efficiency is asymptotically uniform, $\eta_j^{(2),\,FAFM} \to \lambda$ as $j \to \infty$. This uniformity is not a tuned property but a direct consequence of the H^1 norm: the derivative weight ω_j^2 in μ_j exactly matches the acceleration amplification $A_j^{(2)} = \omega_j^2$. No separate hyperparameter is responsible for this match.

The construction extends to higher derivative orders by analogous supervision. This stronger supervision on high-frequency coefficient errors directly suppresses high-frequency artifacts such as action jitter.

D H^∞ Natural Regularity

Proposition 2 ($\hat{v} \in H^\infty$)**.** *Any trajectory decoded by Eq.* (7) *with finite* M *satisfies* $\hat{v} \in H^\infty[0, T_\xi] = \bigcap_{s \geq 0} H^s[0, T_\xi]$*:*

$$\|\hat{v}\|_{H^s}^2 = \frac{T_\xi}{2} \sum_{j=1}^{M} \big(1 + \omega_j^2 + \cdots + \omega_j^{2s}\big) |\hat{c}_j|^2 < \infty, \quad \forall\, s \geq 0.$$

Proof. M, ω_j, and $|\hat{c}_j|$ are finite; a finite sum of finite terms is finite for any s. □ □

Standard chunk FM outputs $[a_0, \ldots, a_{K-1}]$ are discrete sequences; Sobolev regularity is not defined for them. Theorem 1 and Proposition 2 together show that FAFM operates in a Sobolev-regular output space (H^∞, by architecture) and optimizes a Sobolev-consistent objective (H^1, by training).

E Convergence acceleration

For smooth physical trajectories, energy compaction implies $|c_j^*| \lesssim C/j^2$. Under standard L^2 flow matching, the per-mode gradient $2\delta_j$ vanishes for large j, causing slow convergence for fine dynamic details. By Theorem 1, the weight $\mu_j \sim O(j^2)$ acts as a spectral preconditioner that exactly counteracts this $O(1/j^2)$ decay, equalizing gradient pressure across all frequency bands and preventing high-frequency stagnation. This predicts the accelerated convergence observed experimentally.

F Robustness to Embodiment Mechanical Bias

With the DCT-II convention of Eq. (5), a constant offset b applied to the executed sequence propagates to the DCT coefficients as

$$\delta\hat{c}_j\big|_{\text{mech}} = 2b \sum_{n=0}^{K-1} \phi_j(n)\phi_0(n) = 2Kb \cdot \delta_{j0},$$

where $\phi_0(n) \equiv 1$ and the last step follows from discrete cosine orthogonality. Only $\hat{c}_0$ (the DC coefficient, controlling mean position) is perturbed; all $j \geq 1$ shape-defining coefficients are *exactly* unaffected. Under i.i.d. time-varying noise, by contrast, $\mathrm{Var}(\delta\hat{c}_j) = 2K\sigma^2$ uniformly for all j, so no such isolation exists.

Under chunk FM, the same bias b shifts action steps directly: $\delta a_k = b$ for $k = 0, \ldots, K-1$. The perturbation spans the K-dimensional action space, identical in dimensionality to that induced by time-varying noise.

G Settings & Additional Experiments

G.1 Settings

Standalone-policy (small-model) setting. Table 3 summarizes the key optimization and inference settings used in our standalone-policy experiments unless otherwise specified. Across these policy experiments, all methods use the same Transformer backbone with 6 layers, 4 attention heads, and embedding dimension 256. For FAFM, we fix the velocity regularization weight to $\lambda = 1$ throughout, consistent with Eq. (11).

Inference-step budget. For methods that rely on iterative sampling, we use 10 sampling steps at inference for all compared methods. This value is slightly larger than the original setting used by MPD. We adopt the unified budget of 10 because smaller step budgets lead to noticeably degraded performance for several methods, making cross-method comparison less stable.

Execution protocol. All experiments use 10 Hz action execution by default. For obstacle avoidance and LapGym, all methods predict 12 actions and execute the first 5 before replanning every 0.5 s. The VLA experiments follow the OpenPI receding-horizon execution protocol, predicting 50 actions

Table 3: Key optimization and inference settings for the standalone-policy (small-model) experiments.

Setting	Value
Backbone	Transformer
Layers	6
Attention heads	4
Embedding dimension	256
Epochs	3,000 (500 in toy)
Batch size	256
Optimizer	AdamW
Learning rate	1.0×10^{-4}
Betas	$(0.95, 0.999)$
Epsilon	1.0×10^{-8}
Weight decay	1.0×10^{-3}
λ	1
Inference sampling steps	10
Chunk horizon	12

and executing the first 5 before replanning. These matched execution settings facilitate controlled comparisons [51].

VLA setting. Table 4 summarizes the common training settings used for the π_0 and $\pi_{0.5}$ experiments. In both cases, we use our FAFM to train the action expert, and report the fine-tuning strategies for the VLM and action expert separately.

Table 4: Key training settings for the VLA experiments with π_0 and $\pi_{0.5}$.

Setting	Value
Batch size	32
Training steps	30,000
VLM	Full-parameter
Action expert	Full-parameter
Chunk horizon	50

DCT coefficients. We set the number of retained DCT coefficients to $M \approx K/3$, where K is the action chunk horizon. Specifically, $M = 4$ for the standalone-policy experiments ($K = 12$) and $M = 16$ for the VLA experiments ($K = 50$). This choice retains the dominant low-frequency components of manipulation trajectories while suppressing high-frequency noise. This choice is consistent with FAST [17], which applies DCT to robot action tokenization and reports that the analogous compression hyperparameter is insensitive.

For mixed-frequency training, we fix $T_\xi = 5\,\mathrm{s}$ and $M = 16$. Although the source lengths are $K = 25/50/100$ at $5/10/20\,\mathrm{Hz}$, respectively, the retained physical frequency band remains 0-1.6 Hz, since $\nu_j = j/(2T_\xi)$. Thus, FAFM aligns the physical frequency band rather than the compression ratio across recording frequencies. Additional ablation results are provided in Appendix G.2.

G.2 Additional Experiments

DCT coefficient ablation and low-pass filtering. We evaluate the influence of the retained DCT coefficients M on Rope Threading ($K = 12$). We also apply low-pass filtering (LPF) to the FM baseline using the same effective bandwidth as FAFM ($M = 4$).

As shown in Table 5, $M = 4$ achieves the best success rate and smoothness among the tested coefficient counts. FAFM also outperforms the low-pass-filtered FM baseline, indicating that the improvements are not solely attributable to frequency truncation.

Derivative loss weight. We evaluate the sensitivity to the derivative regularization weight λ on Rope Threading.

Table 5: Ablation on DCT coefficients and low-pass filtering on Rope Threading. Results are reported as mean ± std over three training seeds.

Method	SR (%)	LDLJ ↑
FAFM ($M = 1$)	91.00 ± 1.63	-8.07 ± 0.16
FAFM ($M = 2$)	93.00 ± 1.63	-8.48 ± 0.03
FAFM ($M = 4$)	$\mathbf{94.67 \pm 2.62}$	$\mathbf{-7.99 \pm 0.30}$
FAFM ($M = 6$)	94.33 ± 0.94	-8.82 ± 0.32
FAFM ($M = 8$)	89.67 ± 0.47	-9.51 ± 0.11
FM + LPF ($M = 4$)	90.67 ± 1.70	-10.39 ± 0.44

Table 6: Sensitivity to the derivative loss weight on Rope Threading. Results are reported as mean ± std over three training seeds.

λ	SR (%)	LDLJ ↑
0.5	92.33 ± 0.94	-8.02 ± 0.23
1.0	$\mathbf{94.67 \pm 2.62}$	$\mathbf{-7.99 \pm 0.30}$
2.0	91.67 ± 2.36	-8.31 ± 0.03

As shown in Table 6, $\lambda = 1$ achieves the best performance among the tested values.

Mixed-frequency motion smoothness. We further report the motion smoothness of the mixed-frequency baselines in simulation and real-world experiments, corresponding to Fig. 9. As shown in Table 7, FAFM achieves better motion smoothness than both baselines in simulation and real-world experiments.

Table 7: LDLJ under mixed-frequency training. Simulation results are reported over three training seeds. Real-robot results are obtained from one training seed, with standard deviations computed across evaluation episodes.

Method	Sim. LDLJ ↑	Real LDLJ ↑
π_0-Resample	-14.75 ± 0.07	-9.164 ± 3.416
π_0-FreqCond	-14.75 ± 0.10	-8.822 ± 2.254
FAFM	$\mathbf{-13.90 \pm 0.07}$	$\mathbf{-7.84 \pm 1.41}$

Inter-chunk consistency and RTC. FAFM does not explicitly enforce consistency between independently predicted action chunks. We evaluate inter-chunk smoothness using the boundary action jump:

$$D_{\text{boundary}} = \frac{1}{|\mathcal{B}|} \sum_{t \in \mathcal{B}} \|a_t - a_{t-1}\|_2. \tag{25}$$

We further integrate RTC [52] as an inference-time extension of FAFM. At each flow-integration step t, we estimate the final DCT coefficients as

$$\hat{c}_1 = c_t + (1 - t) v_\theta(c_t, o, t), \tag{26}$$

and decode the corresponding action trajectory at the controller timestamps:

$$\hat{A}_1 = D_{\{\tau_i\}}(\hat{c}_1). \tag{27}$$

The RTC consistency objective is

$$\mathcal{L}_{\text{RTC}} = \left\| W \odot \left(Y - D_{\{\tau_i\}}(\hat{c}_1) \right) \right\|_2^2, \tag{28}$$

where Y denotes the aligned overlapping actions from the previous chunk and W is the RTC soft mask. The consistency gradient is propagated through the cosine decoder to guide the coefficient-space flow state. On Rope Threading, the boundary action jump decreases from 1.21 for FM to 0.66 for FAFM, and further to 0.61 for FAFM+RTC.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: By methods and experiments.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Addressed in conclusions.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We have included assumptions and proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Code and details given.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code and details given.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, in Appendix and code.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report mean and standard deviation over three independent training seeds for key mixed-frequency experiments and additional ablations (Fig. 9 and Tables 5-7). We also report episode-level variability for motion smoothness and statistical tests for real-robot experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have included this.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We follow the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Our paper solves problem in robot manipulation, which potentially benefit any robot applications.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Our method improve the capability of robots and is not related to safety issue.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have properly cited these sources respect the license of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We have released the code with structured templates.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: We use LLM for editing only.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.