STABILIZING RL FINE-TUNING OF DIFFUSION VLAS WITH SPARSE MOE ACTION HEADS: ADVANTAGE- AND NOISE-AWARE LOAD BALANCING AND GEOMETRIC-MEAN RATIO

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse mixture-of-experts (MoE) is an effective architecture for scaling vision-language-action (VLA) policies, but imitation learning alone is limited by demonstration coverage and cannot fully exploit MoE capacity under distribution shift. Reinforcement learning (RL) fine-tuning can leverage task feedback, yet directly applying RL to diffusion MoE-VLAs is unstable because continuous denoising dynamics interact with discrete expert routing. We diagnose two failure modes: (1) advantage-signed routing updates sharpen experts for positive-advantage samples and flatten them for negative-advantage samples, while our measurements show larger routing changes at late low-noise steps; and (2) expert switching along the denoising chain causes step-wise likelihood-ratio deviations to accumulate, making the resulting trajectory ratios heavy-tailed and frequently clipped. To address these issues, we propose two minimal interventions. First, as a loss term, Advantage- and Noise-Aware Load Balancing (ANA-LB) applies noise-scaled load balancing only to trajectories with negative group-relative advantage, stabilizing relatively worse samples while preserving positive-advantage expert specialization. Second, for ratio computation, Geometric-Mean Ratio (GMR) uses a geometric-mean surrogate for the accumulated step-wise log-ratios, trading a small amount of exact importance-weighting fidelity for reduced ratio variance. Experiments on LIBERO, CALVIN, RoboTwin 2.0, and real-robot tasks show competitive overall performance, with the clearest gains in several long-horizon settings, while reducing routing collapse, excessive ratio clipping, and training oscillation.

1 INTRODUCTION

Vision-language-action (VLA) policies map multimodal observations to continuous robot control commands Intelligence et al. (2025); Black et al. (2025); Zhao et al. (2023); Kim et al. (2024); Black et al. (2024); Song et al. (2025). Embodied manipulation is inherently heterogeneous: a single policy must coordinate perception, contact-rich control, recovery, and long-horizon temporal composition. These abilities often rely on different visual cues and thinking patterns, making monolithic parameter sharing prone to interference across skills, objects, and task contexts. Sparse mixture-of-experts (MoE) architectures therefore provide a natural scaling mechanism to overcome this multi-task interference. Recent works have successfully integrated MoE into VLA models, demonstrating that routing multimodal observations to specialized experts can effectively mitigate negative transfer across diverse robot morphologies and task clusters Cheng et al. (2025); Yu et al. (2025b); Yang et al. (2025a). By partitioning network capacity to handle distinct manipulation skills, force-feedback patterns, or visual semantics, these approaches establish MoE not merely as a computational scaling tool, but as an architectural match to the modular nature of embodied control. This direction is increasingly relevant to generative action modeling: LingBot-VLA 2.0 replaces the feed-forward layers throughout its action expert with sparse MoE layers to scale action capacity under fixed active computation Wu et al. (2026).

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

However, existing MoE-VLA works predominantly rely on supervised fine-tuning (SFT), which fundamentally limits their potential in dynamic embodied environments. While current MoE architectures successfully scale model capacity on static datasets, they only imitate collected demonstrations and lack interactive corrective feedback. Consequently, current MoE-SFT policies suffer from severe routing degradation under distribution shifts: when facing out-of-distribution states, accumulated errors in long-horizon tasks, or rare contact-rich motions, the static routing learned via imitation often fails to coordinate experts effectively. Instead of exhibiting robust recovery, these MoE policies tend to collapse and overfit to frequent demonstration modes, rendering them brittle in complex real-world executions Din et al. (2025); Firoozi et al. (2025); Kawaharazuka et al. (2025). RL fine-tuning offers a natural path beyond imitation by using task rewards to improve the policy through interaction Guo et al. (2025a); Jaech et al. (2024). Yet directly combining RL with MoE applied to the flow-matching action head is highly non-trivial: the same sparse routing that enables skill specialization also introduces discontinuous expert switching along the multi-step denoising trajectory, creating severe instability in policy-gradient updates.

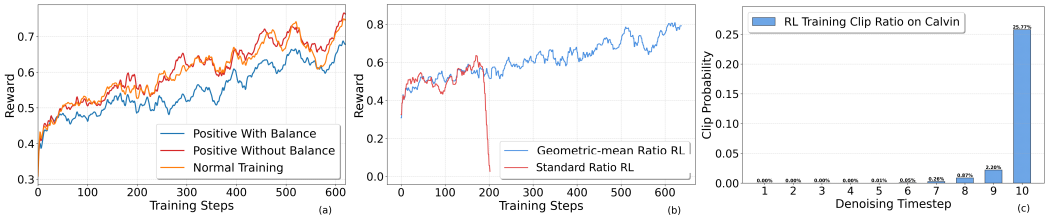


Figure 1: **Empirical Analysis of Instability in MoE RL Fine-Tuning on CALVIN.** (a) Applying load balancing to positive-advantage samples hurts performance, showing local routing instability. (b) The baseline ratio aggregation becomes unstable, while geometric-mean aggregation remains stable, revealing global ratio instability caused by expert switching. (c) The clipping probability increases sharply at later denoising steps, confirming that the instability is concentrated in the late denoising stage.

We identify two fundamental sources of instability in RL fine-tuning diffusion MoE-VLAs. **First, local routing instability**: as shown in Figure 1(a), common load-balancing losses smooth the routing distribution of all samples indiscriminately, which conflicts with advantage-driven policy improvement. Specifically, positive-advantage samples ($A > 0$) sharpen the routing distribution by reinforcing supported experts, whereas negative-advantage samples ($A < 0$) flatten it to encourage exploration (see Theorem 3.1 and Appendix F). Therefore, unconditional load balancing may suppress positive-advantage specialization. The high-noise analysis contracts the responsibility–prior gap; its upper bound becomes less informative at low noise, where our measurements show larger late-step routing changes. **Second, global ratio instability**: as shown in Figure 1(b)(c), expert switching across the denoising chain creates abrupt changes in step-wise likelihood ratios. These deviations accumulate along the denoising chain, producing heavy-tailed trajectory-level ratios that cause excessive clipping and inefficient policy updates.

To address these two issues, we propose two minimally invasive mechanisms. **For the loss term**, we introduce Advantage- and Noise-Aware Load Balancing (ANA-LB). Instead of flattening all routing distributions, ANA-LB applies load balancing only to negative group-relative advantage samples, thereby promoting exploration on relatively worse trajectories without weakening positive-advantage specialization. It scales the regularization strength by a separate ANA-LB weighting schedule $\lambda_{\mathrm{ANA}}(t)$ to balance the auxiliary contribution across steps. **For the ratio computation**, we introduce the Geometric-Mean Ratio (GMR). Rather than following conservative MoE-RL strategies such as IcePop Zhao et al. (2025) that mask high-deviation updates, GMR preserves these trajectory signals while using a biased surrogate to rescale their effective update magnitude. This is important for embodied RL, where low-advantage or boundary rollouts often contain useful corrective information about contact, recovery, precision control, and long-horizon error accumulation. By aggregating per-step likelihood ratios in the log domain and applying geometric averaging across denoising steps, GMR reduces heavy-tailed ratio amplification and clipping frequency as the denoising horizon increases.

We conduct systematic experiments and analyses on LIBERO, CALVIN, RoboTwin 2.0, and real-robot manipulation. The results show competitive overall performance, with the clearest improvements on several long-horizon settings, and verify the expected advantage-signed routing dynamics, more balanced step-wise routing updates, and lower clipping frequency. The direct scope of our method is sparse MoE-VLAs optimized with GRPO-style step-wise likelihood ratios: ANA-LB addresses routing instability specific to the sparse action head, whereas GMR addresses ratio accumulation along the iterative generation path. Based on these findings, **our main contributions are as follows:**

• We identify and theoretically diagnose two instability sources in diffusion MoE-VLA RL fine-tuning: advantage-signed routing updates with low-noise amplification, and heavy-tailed importance ratios induced by expert switching across the denoising chain.

• We propose ANA-LB and GMR as complementary stabilization mechanisms. ANA-LB preserves successful expert specialization by applying noise-aware load balancing only to negative-advantage samples, while GMR retains high-deviation trajectory signals and reduces ratio variance through log-domain geometric averaging.

• We demonstrate competitive performance on LIBERO, CALVIN, RoboTwin 2.0, and real-robot manipulation, with particularly clear gains in several long-horizon settings, while reducing routing collapse, excessive ratio clipping, useful-signal discarding, and oscillatory training behavior.

2 PRELIMINARIES

Flow Matching. Flow Matching learns a time-dependent velocity field $v_\theta(x, \tau)$ and generates actions by integrating an ordinary differential equation (ODE) from noise to data. Given a data action $x^0 \sim p_{\text{data}}$ and a noise action $x^1 \sim p_{\text{noise}}$, we draw $\tau \sim \mathcal{U}(0, 1)$ and construct the linear interpolation $x^\tau = (1 - \tau)x^0 + \tau x^1$. Then, FM is trained by regressing the model velocity field to the target:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{x^0, x^1, \tau}\left[\left\| v_\theta(x^\tau, \tau) - (x^1 - x^0) \right\|_2^2\right]. \tag{1}$$

Starting from $x^{\tau=1} \sim p_{\text{noise}}$, we integrate the ODE backward in time: the denoising trajectory is $x^{\tau_{k-1}} = x^{\tau_k} + \Delta\tau_k\, v_\theta(x^{\tau_k}, \tau_k)$ with $\tau_k = 1 \to \tau_0 = 0$, and x^{τ_0} is returned as the generated action.

GRPO for Diffusion. We view one generation as a trajectory $\gamma = \{\, x_i^T, x_i^{T-1}, \ldots, x_i^0 \,\}$ for the i-th sample. Here, $t = 1, \ldots, T$ indexes the denoising step (superscript), with T being the total number of steps along the trajectory. For each prompt c, we sample a group of G trajectories $\{\gamma_i\}_{i=1}^G$ and obtain rewards R_i. GRPO uses group-relative advantages $A_i = \frac{R_i - \mu_R}{\sigma_R + \delta}$, where $\mu_R = \frac{1}{G}\sum_{j=1}^G R_j$ and $\sigma_R^2 = \frac{1}{G}\sum_{j=1}^G \big(R_j - \mu_R\big)^2$. Let $\pi_\theta(\gamma \mid c)$ denote the trajectory density under the current parameters and $\pi_{\theta_{\text{old}}}$ the behavior policy used to sample trajectories. Define the trajectory ratio $r_i(\theta) = \frac{\pi_\theta(\gamma_i|c)}{\pi_{\theta_{\text{old}}}(\gamma_i|c)}$. The GRPO loss is

$$\mathcal{L}_{\text{GRPO}}(\theta) = \beta\, \mathrm{KL}(\pi_\theta \,\|\, \pi_{\text{ref}}) - \sum\nolimits_{i=1}^G \min\Big(r_i(\theta)A_i,\ \mathrm{clip}\big(r_i(\theta), 1-\epsilon, 1+\epsilon\big)A_i\Big)/G. \tag{2}$$

For diffusion models, a key practical issue is that **ODE sampling is nearly deterministic** beyond the initial noise, which both **limits exploration** and **makes $\pi_\theta(\gamma)$ inconvenient to compute in a stepwise manner**. We therefore use a stochastic differential equation (SDE) sampler with Gaussian reverse transitions (see Appendix C):

$$x_i^{t-1} = x_i^t + m_\theta(x_i^t, t, c) + s(t)\,\varepsilon_{i,t}, \quad \varepsilon_{i,t} \sim \mathcal{N}(0, I). \tag{3}$$

Here m_θ is the score-corrected mean displacement and $s(t)$ is the standard deviation of the Gaussian transition; the exact finite-step definition used in training is given in Appendix C. This gives the Gaussian kernel

$$p_\theta(x_i^{t-1} \mid x_i^t, c) = \mathcal{N}\big(x_i^t + m_\theta(x_i^t, t, c),\ s(t)^2 I\big), \tag{4}$$

thus $\log \pi_\theta(\gamma \mid c) = \sum_{t=1}^T \log p_\theta(x_i^{t-1} \mid x_i^t, c)$, making $r_i(\theta)$ tractable (see Appendix E).

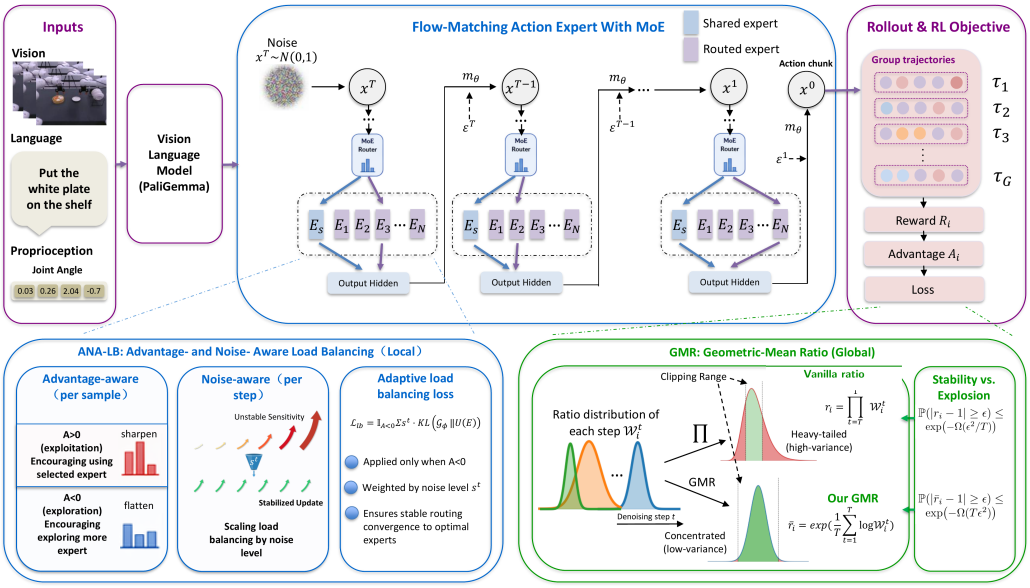


Figure 2: Overview of our flow-based VLA RL fine-tuning framework. We stabilize MoE routing and importance-weight estimation by applying advantage- and noise-aware load balancing and a geometric-mean aggregation of step-wise likelihood ratios.

3 METHOD

Overview. We identify two theoretical issues when fine-tuning MoE-based diffusion VLA policies with RL (Fig. 2). **First**, the advantage sign induces asymmetric routing dynamics during embodied rollouts: positive-advantage robot action trajectories sharpen the expert distribution and reduce routing entropy, which may prematurely over-specialize the policy to a few experts and weaken exploration over alternative manipulation strategies, whereas negative-advantage trajectories naturally flatten the distribution and help the policy recover from relatively worse expert choices. Thus, we apply the load-balancing loss only on trajectories with negative group-relative advantage and weight it adaptively by the step-wise noise coefficient, termed advantage- and noise-aware load balancing (ANA-LB), which regularizes relatively worse trajectories while preserving useful specialization on positive-advantage trajectories. **Second**, diffusion VLA policies generate continuous robot action chunks through long reverse-time denoising chains, where small MoE routing shifts at each denoising step can be amplified into extreme likelihood ratios for the final executed actions. Such ratio volatility is especially harmful in physical-control settings, since small action deviations may lead to different contacts, collisions, or task outcomes. We instead adopt a geometric-mean surrogate ratio (GMR) that aggregates step-wise ratios in the log domain. This power-transformed ratio is not, in general, an unbiased trajectory importance ratio; it changes the exact PPO/GRPO weighting in exchange for reduced ratio variance and fewer excessive clipping events as the denoising horizon increases.

Training pipeline (Algorithm 1). Each iteration rolls out embodied trajectories, computes group-relative advantages, aggregates per-step log-ratios with GMR, and applies one clipped GRPO update. ANA-LB regularizes routing only on negative group-relative advantage samples before the behavior policy is synchronized. The complete loop is given in Appendix 1.

Why this setting introduces additional instability. A flow-matching action policy is naturally sampled by a deterministic ODE, whose intermediate transitions do not provide directly usable likelihoods; the SDE sampler in Eq. equation 3 makes step-wise Gaussian ratios tractable. This does not remove the optimization problem created by the sparse action head: each action chunk requires T denoiser forward passes, and the hard Top-1 router is recomputed at every changing latent state. Expert switches alter subsequent vector fields, latent states, routing decisions, and ratios, with the strongest effect at late low-noise steps and further accumulation under closed-loop execution.

Thus, our method is not a direct transfer of sequence-level geometric averaging from autoregressive RL: ANA-LB stabilizes sparse routing, while GMR controls ratios accumulated through iterative generation. Both mechanisms are used only during fine-tuning and add no deployment-time routing or denoising steps.

3.1 Advantage- and Noise-Aware Load Balancing

We quantify how a MoE load-balancing loss couples with *advantage-signed* policy-gradient updates on router logits and how this coupling changes along SDE denoising steps. To analyze the routing dynamics of the Top-1 MoE used in our implementation, we consider its **differentiable soft-router relaxation**. Under this relaxation, the router produces an expert prior $g_\phi(e \mid x^t)$ over $e \in \{1, \dots, E\}$ given the local router input x^t, where $\phi \subset \theta$ denotes the router parameters as a subset of the full model parameters θ. Let $z_e = z_\phi(x^t)_e$ denote the corresponding *router logit*. Define the expert-conditional likelihood p_θ, i.e., how well expert e explains the result x^{t-1} under the current model. The implemented layer uses *weighted hard Top-1 routing*: only the highest-scoring expert is activated in the forward pass, and its output is multiplied by the corresponding softmax routing probability. The selected gate probability therefore remains in the computation graph. The soft-router result is a local relaxation-based explanation of the selected-expert-versus-router-prior update tendency; the exact hard-routing forward is evaluated separately in our experiments. Related analyses likewise study how gradient-based MoE routing learns expert specialization Kawata et al. (2025). Then the posterior responsibility is

$$q^t(e \mid x^t, x^{t-1}) \triangleq g_\phi(e \mid x^t)\, p_\theta(x^{t-1} \mid x^t, e) / (\sum\nolimits_{j=1}^{E} g_\phi(j \mid x^t)\, p_\theta(x^{t-1} \mid x^t, j)). \tag{5}$$

The responsibility $q^t(e \mid x^t, x^{t-1})$ indicates which expert is most supported by the realized trajectory evidence x^{t-1}. Then, we have:

Theorem 3.1. *Under policy-gradient RL fine-tuning, the router-logit update satisfies*

$$\Delta z_e \;\propto\; A \left(q^t(e \mid x^t, x^{t-1}) - g_\phi(e \mid x^t) \right), \tag{6}$$

where A is the advantage associated with the result x^{t-1}.

Intuitively, $q(\cdot \mid x^t, x^{t-1})$ is the posterior responsibility conditioned on the result (e.g., x^{t-1}), which will decrease uncertainty. Therefore, $q(\cdot \mid x^t, x^{t-1})$ is typically more concentrated than the prior router distribution $g(\cdot \mid x^t)$. Consequently, on **positive-advantage** trajectories ($A > 0$), a "correct" expert tends to have $q_e > g_e$, and Eq. equation 6 increases its logit (hence increasing g_e); conversely, "incorrect" experts satisfy $q_e < g_e$ and are down-weighted. Over RL updates on positive samples, g becomes progressively sharper, improving specialization but reducing exploration across experts.

On **negative group-relative advantage** trajectories ($A < 0$), the sign flips and suppresses the experts associated with the relatively worse outcome. ANA-LB does not treat uniform routing as a terminal objective; it only reduces over-concentration on the current low-advantage choice so that alternative experts can be activated. The task reward still determines which alternative is useful, and once its advantage becomes positive, subsequent policy updates sharpen the router toward that expert. This "flatten-on-negative-advantage, sharpen-on-positive-advantage" interaction motivates the advantage gate in ANA-LB.

Theorem 3.2 (High-noise responsibility contraction)**.** *Under the local Gaussian expert-transition approximation, define the variance scale $\nu^t \triangleq 2(s^t)^2$. When ν^t is large, the responsibility–prior deviation satisfies*

$$q^t(e \mid x^t, x^{t-1}) - g^t(e \mid x^t) = O(1/\nu^t). \tag{7}$$

The analysis above shows that diffusion noise directly controls the sensitivity of expert responsibilities and their router gradients to transition errors. High noise smooths the likelihood differences among experts and thereby contracts the responsibility–prior gap. As the denoising noise decreases, the inverse-variance scaling in the Gaussian transition likelihood magnifies the effect of the same expert residual differences on the responsibilities, leading to stronger router-gradient updates at late low-noise steps. This noise-dependent amplification motivates the step-dependent weighting used by ANA-LB. The complete routing relaxation and noise-dependent analysis are provided in Appendices F and G.

To stabilize learning while preserving beneficial exploration, we make two design choices. First, we apply load balancing only when $A < 0$ to promote exploration on relatively worse trajectories without weakening positive-advantage specialization. Second, we use the step-dependent coefficient $\lambda_{\mathrm{ANA}}(t)$ to down-weight auxiliary updates at low-noise steps and balance their contribution across the denoising chain. For a group $\{x_i^t\}_{i=1}^G$ at denoising step t, our final loss is

$$\mathcal{L}_{\text{final}}(\theta) = \mathcal{L}_{\text{GRPO}}(\theta) + \lambda_{lb}/G \cdot \sum\nolimits_{i=1}^{G} \mathbb{I}_{A_i<0} \sum\nolimits_{t=1}^{T} \lambda_{\mathrm{ANA}}(t) \cdot \mathrm{KL}(g_\phi(\cdot \mid x_i^t) \,\|\, \mathcal{U}(E)). \tag{8}$$

3.2 Geometric-Mean Ratio

GSPO Zheng et al. (2025a) uses a geometric-mean ratio to stabilize autoregressive policy optimization. Our setting has an additional aggregation axis: each executed action chunk is itself generated by multiple stochastic reverse transitions, and hard expert switches perturb both the current transition and the remaining denoising path. We therefore compute a geometric mean within each action chunk and a second geometric mean across the valid chunks of the robot rollout.

Trajectories γ_i are sampled from $\pi_{\theta_{\text{old}}}$, and the target and behavior policies share the same noise prior and variance schedule. For each reverse step, let $w_i^t(\theta)$ denote the ratio of the target and behavior Gaussian transition densities.

Proposition 3.3 (Local variance scaling of the GMR surrogate)**.** *Under a fixed-state, locally Gaussian approximation with conditionally independent step noises, the* geometric-mean surrogate ratio $\bar{r}_i(\theta)$ *is computed as*

$$\bar{r}_i(\theta) \triangleq \exp(\sum\nolimits_{t=1}^{T} \log w_i^t(\theta)/T), \quad \log w_i^t = \langle \varepsilon_i^t,\, \delta m_\theta \rangle / s^t - \|\delta m_\theta\|_2^2/(2(s^t)^2). \tag{9}$$

the centered log-ratio variance scales as $O(1/T)$, whereas the corresponding product-form log-ratio variance scales as $O(T)$. The full conditional statement and its assumptions are given in Appendix H.

Inner-level analysis. The T-step chain directly determines each executed action chunk, so routing-induced ratio fluctuations can alter contacts and end-effector motions. Under the local approximation, Proposition 3.3 gives $O(T)$ product-form versus $O(1/T)$ GMR centered log-ratio variance. This variance reduction suppresses long-chain ratio volatility, and Figure 3(c) empirically verifies the lower clipping frequency.

Outer-level analysis. An embodied trajectory usually contains multiple generated action chunks whose effects accumulate through closed-loop interaction with the environment. Let rollout i contain N_i valid action chunks, each produced by T reverse steps, and define

$$\begin{aligned} w_{ijt}(\theta) &= \frac{p_\theta(x_{ij}^{t-1} \mid x_{ij}^t, c_i)}{p_{\text{old}}(x_{ij}^{t-1} \mid x_{ij}^t, c_i)}, \\ \bar{r}_{ij}(\theta) &= \exp\left(\frac{1}{T}\sum_{t=1}^{T} \log w_{ijt}(\theta)\right), \\ \bar{r}_i(\theta) &= \exp\left(\frac{1}{N_i}\sum_{j=1}^{N_i} \log \bar{r}_{ij}(\theta)\right) = \exp\left(\frac{1}{N_i T}\sum_{j=1}^{N_i}\sum_{t=1}^{T} \log w_{ijt}(\theta)\right). \end{aligned} \tag{10}$$

Here N_i excludes padded chunks and all chunks after termination. Every valid chunk inherits the same trajectory advantage A_i, and PPO-style clipping is applied once to the final trajectory-level ratio $\bar{r}_i$. The inner mean prevents a single reverse transition from dominating its action chunk; the outer mean prevents a single chunk from dominating the rollout update.

The exponent $1/T$ is fixed by the sampler's number of denoising steps rather than tuned from outcomes; the second factor $1/N_i$ averages the valid action chunks in a rollout. The formulation directly applies to randomized flow/diffusion policy optimization with explicit step-wise ratios and a GRPO/PPO-style surrogate. DPPO-style objectives can adopt the same aggregation when such ratios are available, whereas likelihood-free objectives such as FPO require a separate derivation.

4 EXPERIMENTS

Baselines. We compare with Octo Team et al. (2024), OpenVLA Kim et al. (2024), CogACT Li et al. (2024), the π_0 and $\pi_{0.5}$ families, few-shot SFT, and flow-based RL. Flow-SDE/Flow-Noise use dense VLA values reported by πRL Chen et al. (2025a), whereas Ours uses a sparse Top-1 MoE; Table 1 is therefore a cross-architecture reference, while Table 3 provides same-MoE attribution.

Benchmarks. We evaluate three co-equal simulation benchmarks: LIBERO Liu et al. (2023) for compositional manipulation, CALVIN Mees et al. (2022) for long-horizon task chains, and ROBOTWIN 2.0 Chen et al. (2025b) for 50 bimanual tasks with stronger visual variation and domain randomization.

Experimental settings. Simulation experiments use SFT-initialized π_0 and $\pi_{0.5}$ policies, ten denoising steps, group-relative GRPO, binary success rewards, and the same SDE transition; benchmark-specific details and task examples are provided in Appendices I and J. For the real robot, we evaluate 24 tasks with 25 trials per task using two RGB cameras, 7-D proprioception/action control, 30-Hz execution, 50-step action chunks, and a 60-s horizon; the task suite and visual rollouts are given in Appendix K.

4.1 MAIN RESULTS

Table 1: **LIBERO/CALVIN cross-architecture comparison** (success rate %). Dense baselines use dense VLA, whereas Ours uses sparse Top-1 MoE. Flow-based RL values are from πRL; controlled attribution is in Table 3.

Architecture / Backbone	Method	LIBERO (Success Rate %)					CALVIN (Success Rate %)					
		Spatial	Object	Goal	Long	Avg.	1	2	3	4	5	Avg. Len
Published references	Octo	78.9	85.7	84.6	51.1	75.7	77.1	53.5	31.8	20.6	13.6	1.97
	OpenVLA	84.7	88.4	79.2	53.7	76.5	91.3	77.8	62.0	52.1	43.5	3.27
	CogACT	97.2	98.0	90.2	88.8	93.6	83.8	72.9	64.0	55.9	48.0	3.25
	π_0	96.8	98.8	95.8	85.2	94.2	94.7	84.9	74.3	65.2	57.5	3.77
	$\pi_{0.5}$	98.8	98.2	98.0	92.4	96.9	92.7	84.3	76.7	68.8	61.3	3.84
Dense VLA π_0	*SFT*	65.3	64.4	49.8	51.2	57.6	90.3	81.6	72.6	60.9	53.5	3.59
	Flow-SDE	98.4	99.4	96.2	90.2	96.1	96.4	88.0	77.5	70.8	61.7	3.94
	Flow-Noise	99.0	99.2	98.2	93.8	97.6	96.9	88.8	78.0	68.3	59.9	3.92
Sparse Top-1 MoE π_0	**Ours**	99.3	99.4	97.8	94.2	97.7	96.1	88.6	77.2	71.8	62.2	3.95
Dense VLA $\pi_{0.5}$	*SFT*	84.6	95.4	84.6	43.9	77.1	91.2	82.8	71.6	64.4	56.5	3.67
	Flow-SDE	99.6	100.0	98.8	93.0	97.9	99.7	98.2	95.8	91.0	87.0	4.72
	Flow-Noise	99.6	**100.0**	**99.6**	94.0	98.3	99.6	97.6	93.9	89.6	84.5	4.65
Sparse Top-1 MoE $\pi_{0.5}$	**Ours**	**99.7**	99.6	98.9	**95.7**	**98.5**	**99.8**	**98.6**	**96.4**	**93.9**	**89.4**	**4.78**

We separately fine-tune MoE versions of π_0 and $\pi_{0.5}$ on each benchmark; Ours uses few-shot SFT initialization and online RL with binary success rewards under each standard protocol.

Table 1 shows that our sparse MoE method is competitive with the reported dense RL baselines overall, while the exact ordering depends on the backbone and suite. For π_0, our LIBERO average is below Flow-Noise but the CALVIN average chain length is slightly higher than the reported dense baselines. For $\pi_{0.5}$, our method obtains the highest listed averages on both LIBERO and CALVIN, although the average margins over the strongest dense baselines are small. The clearest improvements occur in selected long-horizon settings, including $\pi_{0.5}$ on LIBERO-Long and CALVIN-5.

Table 2: RoboTwin 2.0 success rate (%) on Easy/Hard.

Method	Easy	Hard
ACT	29.7	1.7
Diffusion Policy Chi et al. (2025)	28.0	0.6
π_0	65.9	58.4
X-VLA Zheng et al. (2025b)	72.9	72.8
$\pi_{0.5}$	82.7	76.8
Flow-SDE (π_0, dense)	68.8	63.6
Ours (π_0-MoE)	73.1	70.2
Flow-SDE ($\pi_{0.5}$, dense)	84.2	81.5
Ours ($\pi_{0.5}$-MoE)	**86.6**	**83.5**

Table 2 shows that our method also outperforms the corresponding dense Flow-SDE baseline on RoboTwin 2.0. For π_0, Ours achieves success rates of 73.1 and 70.2 under the Easy and Hard settings, respectively, compared with 68.8 and 63.6 for Flow-SDE, giving gains of 4.3 and 6.6 per-

centage points. For $\pi_{0.5}$, Ours achieves 86.6 and 83.5, compared with 84.2 and 81.5 for Flow-SDE, giving gains of 2.4 and 2.0 points. Compared with the published $\pi_{0.5}$ results, Ours improves by 3.9 points on Easy and 6.7 points on Hard. Different action heads make this an overall comparison; controlled attribution is in the ablations.

4.2 Ablation Studies

We conduct component ablations on the CALVIN benchmark to isolate the functional contribution of each stabilizing component in MoE diffusion-policy RL fine-tuning. Starting from the $\pi_{0.5}$ diffusion VLA policy converted into an MoE architecture, we evaluate vanilla GRPO, standard load balancing, gate-only and noise-only variants, full Advantage- and Noise-Aware Load Balancing (ANA-LB), Geometric-Mean Ratio aggregation (GMR), and their combination.

Table 3: Component ablation on **CALVIN**. ANA-LB and GMR are complementary.

Model	1	2	3	4	5	Avg.Len
$\pi_{0.5}$-MoE	93.5	84.8	77.2	70.5	59.8	3.86
MoE + GRPO	98.7	94.6	89.8	81.2	77.6	4.42
+ Standard LB	99.5	98.3	94.5	88.9	81.5	4.63
+ Gate only	99.6	98.0	93.6	89.8	83.4	4.64
+ Noise only	99.4	97.5	95.3	90.4	84.1	4.67
+ ANA-LB	99.5	97.9	95.0	91.1	86.1	4.70
+ GMR	98.8	96.4	93.1	89.1	85.8	4.63
+ All (**Ours**)	99.8	98.4	96.3	93.6	89.5	4.78

Table 3 details the individual contributions of our stabilization mechanisms on the CALVIN benchmark. Starting from the $\pi_{0.5}$-MoE architecture, vanilla GRPO yields an average chain length of 4.42. Adding standard uniform Load Balancing (+Standard LB) reaches 4.63, but it does not account for the signed exploration dynamics.

The gate-only and noise-only variants reach 4.64 and 4.67, respectively, while combining both choices in ANA-LB reaches 4.70. Independently, replacing the baseline ratio aggregation with our Geometric-Mean Ratio (+GMR) achieves 4.63. Combining ANA-LB and GMR (**Ours**) reaches 4.78. These results show that negative-advantage gating and noise scaling each contribute to routing stabilization, while GMR addresses ratio accumulation; the components are complementary.

4.3 Training Dynamic

This analytical experiment aims to validate, from a training-dynamics perspective, the two mechanisms predicted by our theory and to explain why the proposed routing regularization is effective. First, we test whether the *advantage-signed* router-logit update induces the predicted *flatten-on-negative-advantage, sharpen-on-positive-advantage* behavior, by tracking routing entropy separately on positive-advantage ($A > 0$) and negative-advantage ($A < 0$) trajectories. Second, we examine whether routing changes are larger at late, low-noise denoising steps in the actual SDE implementation. The high-noise analysis provides a contraction result but does not by itself establish a low-noise lower bound; the late-step trend is therefore evaluated empirically. Based on this diagnosis, we further assess whether our two interventions—enabling load balancing only when $A < 0$ to promote exploration on relatively worse trajectories, and using $\lambda_{\text{ANA}}(t)$ to balance per-step contributions—can stabilize learning without sacrificing specialization on $A > 0$ samples.

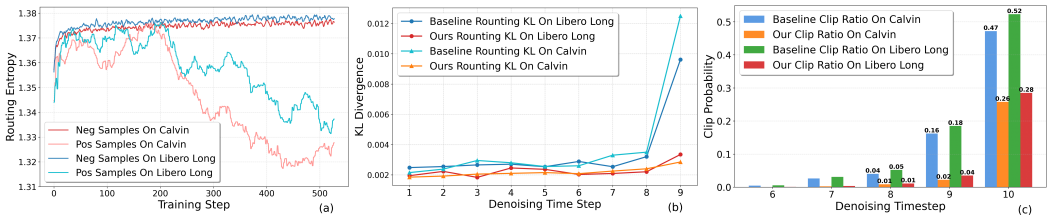


Figure 3: Analytical experiments validating routing and ratio stabilization. **(a)** Advantage-signed routing entropy dynamics. With $A < 0$-gated load balancing, entropy increases more on negative-advantage samples. **(b)** Per-timestep routing-update magnitude (KL divergence). Our intervention suppresses late low-noise dominance. **(c)** Clipping probability as a function of denoising timestep, evaluating ratio stability in diffusion-policy RL.

Figure 3(a) shows the predicted advantage-signed routing dynamics: $A > 0$ sharpens routing, whereas $A < 0$ flattens it, consistent with Theorem 3.1. In Fig. 3(b), we measure the step-local KL divergence between consecutive expert-selection distributions. Baseline updates peak at late low-noise steps, while $A < 0$-gated, $\lambda_{\mathrm{ANA}}(t)$-weighted balancing suppresses this peak and equalizes contributions, as predicted by Theorem 3.2. Together, the plots explain how ANA-LB stabilizes routing without removing success-driven specialization.

4.4 Clipping Frequency

We evaluate clipping probability across denoising timesteps on CALVIN and LIBERO-Long under identical architectures, training settings, and clipping thresholds. Figure 3(c) shows that clipping increases with the denoising horizon for the baseline, whereas GMR remains consistently lower and the gap widens at longer horizons. This supports its ability to reduce ratio explosion and preserve more reliable updates.

4.5 Real-World Robot Evaluation

Experimental Setup. We deployed the policy on a Piper robotic arm and evaluated 24 manipulation tasks spanning six capability dimensions (4 tasks per category, 25 trials per task). Thus, each category aggregates 100 trials and each method is evaluated on 600 trials. We compare a sparse $\pi_{0.5}$-MoE SFT initialization, its vanilla Flow-SDE/GRPO RL baseline, and our method under the same sampler, reward, and evaluation protocol.

Table 4: Real-world evaluation on the Piper arm. Each category contains 100 trials per method.

Method	Basic	Prec.	Long	Dist.	Occl.	Mat. Gen.	Avg.
$\pi_{0.5}$-MoE SFT	89.0	73.0	69.0	66.0	52.0	54.0	67.2
+ Flow-SDE	93.0	68.0	67.0	70.0	65.0	68.0	71.8
+ Ours	**94.0**	**79.0**	**80.0**	**77.0**	**71.0**	**69.0**	**78.3**

Results. Table 4 separates the benefit of RL from that of our stabilizers. Vanilla Flow-SDE/GRPO improves average success over SFT by about 4.7 points (71.8 vs. 67.2); under the matched MoE setup, ANA-LB+GMR adds a further 6.5 points (78.3). The largest gains over vanilla RL occur in Precision, Long-Horizon, and Disturbance (11, 13, and 7 points), where precise correction, multi-stage execution, and recovery are most demanding.

5 Limitations and Future Work

Likelihood-free objectives provide a complementary direction for future work. DiffusionNFT trains from clean terminal samples Zheng et al. (2026), while π-StepNFT adds step-wise supervision for flow-based VLAs Wang et al. (2026). Their ratio-free formulations suggest a natural next step: applying NFT-style objectives to sparse MoE-VLA policies and studying how advantage-conditioned routing and expert switching behave when policy updates no longer rely on likelihood ratios. We will investigate MoE-RL training with these objectives, together with routing-aware regularization and long-horizon evaluation, to combine the broader supervision choices of likelihood-free RL with the specialization benefits of sparse action experts.

6 Conclusion

We studied RL fine-tuning of diffusion MoE-VLA policies and identified two coupled instability sources: advantage-conditioned hard routing at each denoising state and ratio accumulation across the reverse chain. These observations motivate ANA-LB, which regularizes only negative-advantage routes with noise-dependent strength, and GMR, which geometrically aggregates step-wise and action-chunk ratios. Across the evaluated simulated and real-robot settings, the method is competitive overall, shows its clearest gains on several long-horizon tasks, and reduces routing oscillation and excessive clipping. The results support routing-aware and generation-aware stabilization as a practical direction for sparse generative robot policies.

AI USE STATEMENT

The authors used language-model assistance for translation, grammar editing, and document-formatting support. All technical claims, methods, experimental settings, numerical results, and final interpretations are the responsibility of the authors and will be independently verified before submission.

REFERENCES

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

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

Kevin Black, Manuel Y Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. *arXiv preprint arXiv:2506.07339*, 2025.

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.

Kang Chen, Zhihao Liu, Tonghe Zhang, Zhen Guo, Si Xu, Hao Lin, Hongzhi Zang, Xiang Li, Quanlu Zhang, Zhaofei Yu, et al. π_{RL}: Online rl fine-tuning for flow-based vision-language-action models. *arXiv preprint arXiv:2510.25889*, 2025a.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025b.

Baiye Cheng, Tianhai Liang, Suning Huang, Maanping Shao, Feihong Zhang, Botian Xu, Zhengrong Xue, and Huazhe Xu. Moe-dp: An moe-enhanced diffusion policy for robust long-horizon robotic manipulation with skill decomposition and failure recovery. *arXiv preprint arXiv:2511.05007*, 2025.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Yu Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, 2024.

Muhayy Ud Din, Waseem Akram, Lyes Saad Saoud, Jan Rosell, and Irfan Hussain. Vision language action models in robotic manipulation: A systematic review. *arXiv preprint arXiv:2507.10672*, 2025.

Zhiying Du, Bei Liu, Yaobo Liang, Yichao Shen, Haidong Cao, Xiangyu Zheng, Zhiyuan Feng, Zuxuan Wu, Jiaolong Yang, and Yu-Gang Jiang. Himoe-vla: Hierarchical mixture-of-experts for generalist vision-language-action policies. *arXiv preprint arXiv:2512.05693*, 2025.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Roya Firoozi, Johnathan Tucker, Stephen Tian, Anirudha Majumdar, Jiankai Sun, Weiyu Liu, Yuke Zhu, Shuran Song, Ashish Kapoor, Karol Hausman, et al. Foundation models in robotics: Applications, challenges, and the future. *The International Journal of Robotics Research*, 44(5): 701–739, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025a.

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

Xiaoxuan He, Siming Fu, Yuke Zhao, Wanli Li, Jian Yang, Dacheng Yin, Fengyun Rao, and Bo Zhang. Tempflow-grpo: When timing matters for grpo in flow models. *arXiv preprint arXiv:2508.04324*, 2025.

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.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Kento Kawaharazuka, Jihoon Oh, Jun Yamada, Ingmar Posner, and Yuke Zhu. Vision-language-action models for robotics: A review towards real-world applications. *IEEE Access*, 2025.

Ryotaro Kawata, Kohsei Matsutani, Yuri Kinoshita, Naoki Nishikawa, and Taiji Suzuki. Mixture of experts provably detect and learn the latent cluster structure in gradient-based learning, 2025.

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.

Wouter Kool, Herke Van Hoof, and Max Welling. Attention, learn to solve routing problems! *arXiv preprint arXiv:1803.08475*, 2018.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.

Haozhan Li, Yuxin Zuo, Jiale Yu, Yuhao Zhang, Zhaohui Yang, Kaiyan Zhang, Xuekai Zhu, Yuchen Zhang, Tianxing Chen, Ganqu Cui, et al. Simplevla-rl: Scaling vla training via reinforcement learning. *arXiv preprint arXiv:2509.09674*, 2025a.

Junzhe Li, Yutao Cui, Tao Huang, Yinping Ma, Chun Fan, Miles Yang, and Zhao Zhong. Mixgrpo: Unlocking flow-based grpo efficiency with mixed ode-sde. *arXiv preprint arXiv:2507.21802*, 2025b.

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.

Yunfei Li, Xiao Ma, Jiafeng Xu, Yu Cui, Zhongren Cui, Zhigang Han, Liqun Huang, Tao Kong, Yuxiao Liu, Hao Niu, et al. Gr-rl: Going dexterous and precise for long-horizon robotic manipulation. *arXiv preprint arXiv:2512.01801*, 2025c.

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

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

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025b.

Jijia Liu, Feng Gao, Bingwen Wei, Xinlei Chen, Qingmin Liao, Yi Wu, Chao Yu, and Yu Wang. What can rl bring to vla generalization? an empirical study. *arXiv preprint arXiv:2505.19789*, 2025c.

Guanxing Lu, Wenkai Guo, Chubin Zhang, Yuheng Zhou, Haonan Jiang, Zifeng Gao, Yansong Tang, and Ziwei Wang. Vla-rl: Towards masterful and general robotic manipulation with scalable reinforcement learning. *arXiv preprint arXiv:2505.18719*, 2025.

Max Sobol Mark, Tian Gao, Georgia Gabriela Sampaio, Mohan Kumar Srirama, Archit Sharma, Chelsea Finn, and Aviral Kumar. Policy agnostic rl: Offline rl and online rl fine-tuning of any class and backbone. *arXiv preprint arXiv:2412.06685*, 2024.

David McAllister, Songwei Ge, Brent Yi, Chung Min Kim, Ethan Weber, Hongsuk Choi, Haiwen Feng, and Angjoo Kanazawa. Flow matching policy gradients. *arXiv preprint arXiv:2507.21053*, 2025.

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

Atharva Mete, Haotian Xue, Albert Wilcox, Yongxin Chen, and Animesh Garg. Quest: Self-supervised skill abstractions for learning continuous control. *Advances in Neural Information Processing Systems*, 37:4062–4089, 2024.

Seohong Park, Qiyang Li, and Sergey Levine. Flow q-learning. *arXiv preprint arXiv:2502.02538*, 2025.

Samuel Pfrommer, Yixiao Huang, and Somayeh Sojoudi. Reinforcement learning for flow-matching policies. *arXiv preprint arXiv:2507.15073*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Allen Z Ren, Justin Lidard, Lars L Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. *arXiv preprint arXiv:2409.00588*, 2024.

Weijie Shen, Yitian Liu, Yuhao Wu, Zhixuan Liang, Sijia Gu, Dehui Wang, Tian Nian, Lei Xu, Yusen Qin, Jiangmiao Pang, et al. Expertise need not monopolize: Action-specialized mixture of experts for vision-language-action learning. *arXiv preprint arXiv:2510.14300*, 2025.

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

Wenxuan Song, Jiayi Chen, Pengxiang Ding, Han Zhao, Wei Zhao, Zhide Zhong, Zongyuan Ge, Jun Ma, and Haoang Li. Accelerating vision-language-action model integrated with action chunking via parallel decoding. *arXiv preprint arXiv:2503.02310*, 2025.

Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive post-training for vision-language-action models. *arXiv preprint arXiv:2505.17016*, 2025.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Andrew Wagenmaker, Mitsuhiko Nakamoto, Yunchu Zhang, Seohong Park, Waleed Yagoub, Anusha Nagabandi, Abhishek Gupta, and Sergey Levine. Steering your diffusion policy with latent space reinforcement learning. *arXiv preprint arXiv:2506.15799*, 2025.

Siting Wang, Xiaofeng Wang, Zheng Zhu, Minnan Pei, Xinyu Cui, Cheng Deng, Jian Zhao, Guan Huang, Haifeng Zhang, and Jun Wang. π-StepNFT: Wider space needs finer steps in online rl for flow-based vlas. *arXiv preprint arXiv:2603.02083*, 2026.

Junjie Wen, Yichen Zhu, Minjie Zhu, Zhibin Tang, Jinming Li, Zhongyi Zhou, Xiaoyu Liu, Chaomin Shen, Yaxin Peng, and Feifei Feng. Diffusionvla: Scaling robot foundation models via unified diffusion and autoregression. In *Forty-second International Conference on Machine Learning*, 2025.

Hongjie Wu, Mingqin Zhang, Linchao He, Ji-Zhe Zhou, and Jiancheng Lv. Enhancing diffusion model stability for image restoration via gradient management. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 10768–10777, 2025.

Wei Wu, Fangjing Wang, Fan Lu, He Sun, Shi Liu, Yunnan Wang, Yibin Yan, Yong Wang, Shuailei Ma, Xinyang Wang, et al. From foundation to application: Improving vla models in practice. *arXiv preprint arXiv:2607.06403*, 2026.

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

Jiange Yang, Haoyi Zhu, Yating Wang, Gangshan Wu, Tong He, and Limin Wang. Tra-moe: Learning trajectory prediction model from multiple domains for adaptive policy conditioning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 6960–6970, 2025a.

Zhenjie Yang, Yilin Chai, Xiaosong Jia, Qifeng Li, Yuqian Shao, Xuekai Zhu, Haisheng Su, and Junchi Yan. Drivemoe: Mixture-of-experts for vision-language-action model in end-to-end autonomous driving. *arXiv preprint arXiv:2505.16278*, 2025b.

Chao Yu, Yuanqing Wang, Zhen Guo, Hao Lin, Si Xu, Hongzhi Zang, Quanlu Zhang, Yongji Wu, Chunyang Zhu, Junhao Hu, et al. Rlinf: Flexible and efficient large-scale reinforcement learning via macro-to-micro flow transformation. *arXiv preprint arXiv:2509.15965*, 2025a.

Jiawen Yu, Hairuo Liu, Qiaojun Yu, Jieji Ren, Ce Hao, Haitong Ding, Guangyu Huang, Guofan Huang, Yan Song, Panpan Cai, et al. Forcevla: Enhancing vla models with a force-aware moe for contact-rich manipulation. *arXiv preprint arXiv:2505.22159*, 2025b.

Hongzhi Zang, Mingjie Wei, Si Xu, Yongji Wu, Zhen Guo, Yuanqing Wang, Hao Lin, Liangzhi Shi, Yuqing Xie, Zhexuan Xu, et al. Rlinf-vla: A unified and efficient framework for vla+ rl training. *arXiv preprint arXiv:2510.06710*, 2025.

Tonghe Zhang, Chao Yu, Sichang Su, and Yu Wang. Reinflow: Fine-tuning flow matching policy with online reinforcement learning. *arXiv preprint arXiv:2505.22094*, 2025.

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.

Xin Zhao, Yongkang Liu, Kuan Xu, Jia Guo, Zihao Wang, Yan Sun, Xinyu Kong, Qianggang Cao, Liang Jiang, Zujie Wen, et al. Small leak can sink a great ship–boost rl training on moe with icepop!, sep 2025. *URL https://ringtech. notion. site/icepop*, 2025.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025a.

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*, 2025b.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. Diffusionnft: Online diffusion reinforcement with forward process. In *International Conference on Learning Representations*, volume 2026, pp. 134129–134150, 2026.

Zewei Zhou, Tianhui Cai, Seth Z Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. *arXiv preprint arXiv:2506.13757*, 2025.

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*, pp. 2165–2183. PMLR, 2023.

A Training Algorithm

Algorithm 1 MoE-Diffusion RL Fine-Tuning with ANA-LB and GMR

Require: Initial behavior policy $\pi_{\theta_{\text{old}}}$, learning rate η, load-balancing coefficient λ_{lb}

while not converged **do**
  Rollout: Generate G rollouts; rollout i contains N_i valid action chunks
  Evaluate: Execute chunks to obtain a trajectory reward R_i and advantage A_i
  Advantage: Compute group-relative advantages $A_i = (R_i - \mu_R)/(\sigma_R + \delta)$
  for each trajectory $i \in \{1, \dots, G\}$ **do**
    for each valid action chunk $j \in \{1, \dots, N_i\}$ **do**
      for each denoising step $t = T, T-1, \dots, 1$ **do**
        Compute the step-wise log-ratio $\log w_{ijt}(\theta)$
      end for
      Compute $\bar{r}_{ij} = \exp\big(T^{-1} \sum_t \log w_{ijt}\big)$
    end for
    Global Intervention: Compute $\bar{r}_i = \exp\big(N_i^{-1} \sum_j \log \bar{r}_{ij}\big)$
  end for
  Apply clipping once to each trajectory-level $\bar{r}_i$ in the GRPO surrogate
  Local Intervention: Compute ANA-LB loss and RL loss using Eq. (8)
  Update: Update parameters $\theta \leftarrow \theta - \eta \nabla_\theta \big(\mathcal{L}_{\text{GRPO}}(\theta) + \mathcal{L}_{\text{lb}}(\theta)\big)$
  Sync: Update behavior policy $\theta_{\text{old}} \leftarrow \theta$
end while

B Related Work

Generative Policies for VLA. Early VLA systems predominantly follow an *autoregressive* paradigm that discretizes control into token-like representations, enabling language-conditioned action generation but inevitably constraining the resolution needed for fine-grained and contact-rich manipulation Zitkovich et al. (2023); Zhou et al. (2025); Kim et al. (2024). Recent progress in generalist robot policies demonstrates that scaling vision-language backbones together with such discrete decoders can yield broad task coverage, including representative systems Team et al. (2024) and RT Brohan et al. (2022). Motivated by the need for more *dexterous* and *continuous* behaviors, a growing line of work adopts *generative* decoding for actions, where diffusion/flow-style iterative transport maps Gaussian noise to *action chunks* conditioned on multimodal observations Chi et al. (2025); Liu et al. (2025a); Wen et al. (2025); Xiang et al. (2026). This generative formulation naturally models continuous, multi-modal action distributions and supports higher-frequency chunked control, leading to stronger physical dexterity Shukor et al. (2025); Intelligence et al. (2025); recent large-scale systems further exemplify this direction, e.g., GR00T Bjorck et al. (2025). These trends motivate studying how to further improve large generative VLAs beyond SFT, especially via online interaction and reinforcement learning.

VLA via Mixture-of-Experts. Scaling VLA models raises both capacity and interference challenges, motivating Mixture-of-Experts (MoE) Fedus et al. (2022); Lepikhin et al. (2020); Dai et al. (2024) as a principled way to expand representational power. Specifically, real-world robotic datasets suffer from extreme hardware heterogeneity—including varying degrees of freedom, camera configurations, and control frequencies Yang et al. (2025a); Du et al. (2025)—as well as long-tailed task distributions. Furthermore, manipulation tasks encompass diverse behavioral primitives, such as approaching, grasping, and releasing Shen et al. (2025). Recent MoE-VLA designs explore hierarchical expert organizations Du et al. (2025), adaptive routing Shen et al. (2025), and expert aggregation Yu et al. (2025b), aiming to better handle multi-task heterogeneity and reduce negative transfer. MoE has been actively developed alongside *autoregressive* VLA backbones to improve scaling efficiency and specialization Yang et al. (2025a); Yu et al. (2025b); more recently, MoE ideas have also started to interface with *generative* action modeling by introducing action-specialized experts or expertized decoders for continuous control Yang et al. (2025b). Despite these architectural advances, existing MoE-VLAs exclusively rely on offline SFT, leaving the critical challenge of online RL fine-tuning completely unexplored. However, MoE routing introduces additional non-stationarity during fine-tuning: small gating shifts can substantially change the *effective* policy,

which becomes particularly problematic under on-policy RL updates Zhao et al. (2025), and calls for routing-aware stabilization when combining MoE with online RL.

RL Training Stability. Online RL fine-tuning for VLA models is notoriously sensitive to optimization instability, including high-variance policy-gradient updates, non-stationarity from frequent policy changes, and brittleness under sparse or long-horizon rewards Ren et al. (2024); Pfrommer et al. (2025); Park et al. (2025); Wu et al. (2025). Recent VLA-focused RL studies show that online RL can substantially improve generalization and long-horizon behaviors, e.g., SimpleVLA-RL Li et al. (2025a), RL4VLA Liu et al. (2025c) with comparisons among PPO/GRPO and preference-style objectives such as DPO Rafailov et al. (2023), as well as VLA-RL Lu et al. (2025), iRe-VLA Guo et al. (2025b), RIPT-VLA Tan et al. (2025) built on RLOO Kool et al. (2018) and QueST Mete et al. (2024), and scalable training frameworks such as RLinf-VLA Yu et al. (2025a); Zang et al. (2025). However, extending these advances to *flow-based* or diffusion-style policies introduces an additional core difficulty: *tractable ratio estimation* along iterative generation trajectories Li et al. (2025c). Recent flow-RL methods address this issue through different mechanisms, including ODE-to-SDE conversion for stochastic exploration in Flow-GRPO Liu et al. (2025b), hybrid rollouts in Mix-GRPO Li et al. (2025b) and TempFlow-GRPO He et al. (2025), learnable-noise and discrete-time reformulations in ReinFlow Zhang et al. (2025), objective reframing in FPO McAllister et al. (2025), critic-to-policy distillation in PA-RL Mark et al. (2024), and latent-noise RL in DSRL Wagenmaker et al. (2025). These stability issues are further amplified in MoE-based generative policies: routing drift can abruptly change the effective policy, increasing ratio variance and undermining on-policy assumptions, thereby motivating trajectory-aware (yet step-sensitive) ratio control together with routing-aware regularization for stable RL fine-tuning.

C From ODE Sampling to SDE Sampling

C.1 Why the ODE sampler is problematic for policy-gradient style objectives

In Flow Matching, generation is commonly performed by integrating a reverse-time ODE

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = v_\theta(x(t), t, c), \qquad t : 1 \to 0, \tag{11}$$

with initial condition $x(1) \sim p_{\text{noise}}$. In practice, we discretize time $1 = t_\tau > t_{\tau-1} > \cdots > t_0 = 0$ and apply an explicit solver (e.g., Euler). Let the discrete states be denoted $x^{(\tau)}$ at time t_τ. Then the Euler step reads

$$x^{(\tau-1)} = x^{(\tau)} + \Delta t_\tau \, v_\theta(x^{(\tau)}, t_\tau, c), \qquad \Delta t_\tau := t_{\tau-1} - t_\tau < 0. \tag{12}$$

Define the deterministic displacement

$$m_\theta(x^{(\tau)}, t_\tau, c) := \Delta t_\tau \, v_\theta(x^{(\tau)}, t_\tau, c). \tag{13}$$

A key issue for reinforcement-learning style objectives is that the induced transition kernel is *degenerate*. Specifically, equation 12 implies a Dirac transition

$$p_\theta^{\text{ODE}}(x^{(\tau-1)} \mid x^{(\tau)}, c) = \delta\big(x^{(\tau-1)} - x^{(\tau)} - m_\theta(x^{(\tau)}, t_\tau, c)\big), \tag{14}$$

so the trajectory distribution $\pi_\theta(\tau \mid c)$ is supported on a measure-zero manifold determined by the initial noise and the deterministic integrator. Consequently:

- **Exploration is severely limited:** once $x^{(T)}$ is sampled, all subsequent states are deterministic under θ.
- **Stepwise likelihood is inconvenient:** $\log p_\theta^{\text{ODE}}$ is not a regular density w.r.t. Lebesgue measure (it is a distribution), hence the usual log-likelihood ratio $\log \pi_\theta(\tau \mid c) - \log \pi_{\theta_{\text{old}}}(\tau \mid c)$ is not well-defined in the standard sense.

These issues motivate a *stochastic relaxation* of the sampler: we inject a small amount of diffusion noise so that each step admits a proper density and exploration is enabled.

C.2 Discrete SDE sampler used in training

The experiments use a score-corrected reverse-time SDE rather than a bare ODE drift with independent noise. For the linear interpolation path used by flow matching, the velocity network gives the local score estimate

$$\nabla_x \widehat{\log p_t(x \mid c)} = -\frac{x}{t} - \frac{1-t}{t} v_\theta(x, t, c). \tag{15}$$

Using this estimate, the reverse drift is

$$b_\theta(x, t, c) = v_\theta(x, t, c) - \frac{\sigma(t)^2}{2} \nabla_x \widehat{\log p_t(x \mid c)} = v_\theta(x, t, c) + \frac{\sigma(t)^2}{2t}\big(x + (1-t) v_\theta(x, t, c)\big). \tag{16}$$

Using the idealized reverse-time interval $1 = t_K > t_{K-1} > \cdots > t_0 = 0$, with $\Delta t_\tau = t_{\tau-1} - t_\tau < 0$, Euler–Maruyama gives

$$x^{(\tau-1)} = x^{(\tau)} + \Delta t_\tau b_\theta(x^{(\tau)}, t_\tau, c) + \sigma(t_\tau)\sqrt{|\Delta t_\tau|}\, \varepsilon_\tau, \qquad \varepsilon_\tau \sim \mathcal{N}(0, I), \tag{17}$$

so the discrete transition standard deviation is

$$s_\tau = \sigma(t_\tau)\sqrt{|\Delta t_\tau|}. \tag{18}$$

Here s_τ denotes the transition standard deviation used in the likelihood ratio; it is distinct from the variance scale $\nu_\tau = 2s_\tau^2$ used in Appendix G and from the ANA-LB coefficient $\lambda_{\mathrm{ANA}}(t)$. The implementation uses ten finite denoising steps and the prescribed noise schedule with endpoint handling at the limiting values. The idealized formulas are not evaluated at a singular endpoint; the finite grid and endpoint values are supplied by the sampler configuration. Each transition therefore has a tractable Gaussian density used for the step-wise log-ratio.

C.3 Interpretation and limiting case

The sampler is a stochastic relaxation of the deterministic ODE integrator:

- As $\sigma(t_\tau) \to 0$, the Gaussian transition collapses to the Dirac kernel equation 14, and the sampler reduces to the ODE trajectory.
- For nonzero $\sigma(t_\tau)$, the transition has an absolutely continuous density, which enables exploration and tractable step-wise log-densities.

Replacing deterministic ODE transitions by equation 17 therefore enables exploration beyond the initial noise and makes $\log \pi_\theta(\tau \mid c)$ computable via the Gaussian factorization equation 46.

D Theoretical Guarantee

D.1 Marginal-preserving SDE and probability-flow equivalence

The preceding section describes the finite-step sampler used for tractable policy updates. Here we give an *idealized continuous-time construction* whose probability flow has the same marginals as the original ODE. This result is a theoretical interpretation of the score-corrected drift, not an exact claim about the learned finite-step implementation.

Probability flow of the ODE. Let $p_t(x)$ denote the marginal density of $x(t)$ under the reverse-time ODE

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = v_\theta(x(t), t, c). \tag{19}$$

Assuming sufficient regularity (e.g., v_θ is locally Lipschitz in x and continuously differentiable), the density evolution satisfies the continuity (Liouville) equation:

$$\partial_t p_t(x) = -\nabla \cdot \big(v_\theta(x, t, c)\, p_t(x)\big). \tag{20}$$

A marginal-preserving forward SDE. Consider a forward SDE (in the same time direction) with scalar diffusion coefficient $\sigma(t) \geq 0$:

$$\mathrm{d}x(t) = f_\theta(x(t), t, c)\, \mathrm{d}t + \sigma(t)\, \mathrm{d}W(t). \tag{21}$$

Its marginal density satisfies the Fokker–Planck equation

$$\partial_t p_t(x) = -\nabla \cdot \big(f_\theta(x,t,c)\, p_t(x)\big) + \frac{1}{2}\nabla^2\big(\sigma(t)^2\, p_t(x)\big), \tag{22}$$

where ∇^2 denotes the Laplacian.

When $\sigma(t)$ is scalar and independent of x, we have $\nabla^2(\sigma(t)^2 p_t) = \sigma(t)^2\nabla^2 p_t = \nabla\cdot\big(\sigma(t)^2\nabla p_t\big)$. Therefore, choosing the drift as

$$f_\theta(x,t,c) = v_\theta(x,t,c) + \frac{\sigma(t)^2}{2}\,\nabla \log p_t(x) \tag{23}$$

makes equation 22 reduce exactly to equation 20:

$$\begin{aligned} -\nabla\cdot\big(f_\theta p_t\big) + \frac{1}{2}\sigma(t)^2\nabla^2 p_t &= -\nabla\cdot\big(v_\theta p_t\big) - \nabla\cdot\left(\frac{\sigma(t)^2}{2}\nabla\log p_t\, p_t\right) + \frac{1}{2}\sigma(t)^2\nabla^2 p_t \\ &= -\nabla\cdot\big(v_\theta p_t\big) - \frac{\sigma(t)^2}{2}\nabla\cdot(\nabla p_t) + \frac{1}{2}\sigma(t)^2\nabla^2 p_t \\ &= -\nabla\cdot\big(v_\theta p_t\big). \end{aligned} \tag{24}$$

Hence, the SDE equation 21 with drift equation 23 shares the *same marginals* $\{p_t\}_{t\in[0,1]}$ as the ODE, for any chosen noise schedule $\sigma(t)$.

Reverse-time SDE and a practical sampler. For reinforcement learning, we need a reverse-time sampler that starts from noise at $t = 1$ and generates $x(0)$. The reverse-time dynamics corresponding to equation 21 is (under standard regularity conditions):

$$\mathrm{d}x(t) = \Big(f_\theta(x(t),t,c) - \sigma(t)^2\nabla\log p_t(x(t))\Big)\mathrm{d}t + \sigma(t)\,\mathrm{d}\bar{W}(t), \qquad t: 1\to 0, \tag{25}$$

where $\bar{W}(t)$ denotes a Wiener process in reverse time. Substituting equation 23 into equation 25, we obtain

$$\mathrm{d}x(t) = \left(v_\theta(x(t),t,c) - \frac{\sigma(t)^2}{2}\,\nabla\log p_t(x(t))\right)\mathrm{d}t + \sigma(t)\,\mathrm{d}\bar{W}(t), \qquad t: 1\to 0. \tag{26}$$

Equation equation 26 is an idealized *marginal-preserving reverse-time SDE* whose probability flow matches that of the original ODE. In practice, we use a learned velocity, an approximate score identity, a finite denoising grid, and endpoint truncation; the resulting sampler is therefore only an approximation to this continuous-time construction.

Connection to our discrete sampler and convergence. To discretize equation 26, we introduce a grid $1 = t_K > t_{K-1} > \cdots > t_0 = 0$ and denote discrete states by $x^{(\tau)}$ at time t_τ, where $\tau = K, K-1, \ldots, 0$ is the discrete denoising step index (superscript). Applying Euler–Maruyama gives

$$x^{(\tau-1)} = x^{(\tau)} + \Delta t_\tau\left(v_\theta(x^{(\tau)},t_\tau,c) - \frac{\sigma(t_\tau)^2}{2}\,\widehat{\nabla\log p_{t_\tau}}(x^{(\tau)})\right) + \sigma(t_\tau)\sqrt{|\Delta t_\tau|}\,\varepsilon_\tau, \tag{27}$$

with $\Delta t_\tau := t_{\tau-1} - t_\tau < 0$ and $\varepsilon_\tau \sim \mathcal{N}(0, I)$. Under standard conditions for Euler–Maruyama (e.g., Lipschitz drift and linear growth), as the maximum step size $\max_\tau |\Delta t_\tau| \to 0$ and the score and velocity estimates become exact, the discrete-time process converges weakly to the continuous-time SDE. The finite-step learned sampler used in our experiments should therefore be understood as a stochastic approximation motivated by this construction, rather than as exactly marginal-preserving. It nevertheless retains tractable step-wise likelihoods for policy-gradient objectives.

D.2 How to sample the marginal-preserving SDE with only a velocity network

Section D.1 shows that there exists a *marginal-preserving* reverse-time SDE whose probability flow matches the ODE, but its drift involves the intractable score $\nabla_x \log p_t(x \mid c)$. This section shows that, for the *linear interpolation path* used in flow matching, the score can be expressed in closed form using the *velocity field* alone. Therefore, sampling requires only a single velocity network v_θ.

Setup. Flow matching defines the interpolation

$$x_t = (1-t)x_0 + tx_1, \qquad x_0 \sim p_{\text{data}}(\cdot \mid c), \;\; x_1 \sim \mathcal{N}(0, I), \qquad t \in (0, 1], \tag{28}$$

and the optimal velocity satisfies

$$v^*(x, t, c) = \mathbb{E}[x_1 - x_0 \mid x_t = x, c]. \tag{29}$$

In practice, v_θ is trained to approximate v^*.

Lemma (score–velocity identity for the FM path). Let $p_t(x \mid c)$ denote the marginal density of x_t induced by equation 28. Under mild regularity conditions, the score can be written as

$$\nabla_x \log p_t(x \mid c) = -\frac{1}{t}\, x - \frac{1-t}{t}\, v^*(x, t, c). \tag{30}$$

Proof. First, using the definition of v^*,

$$v^*(x, t, c) = \mathbb{E}[x_1 \mid x_t = x, c] - \mathbb{E}[x_0 \mid x_t = x, c]. \tag{31}$$

Taking conditional expectation on both sides of equation 28 yields

$$x = (1-t)\,\mathbb{E}[x_0 \mid x_t = x, c] + t\,\mathbb{E}[x_1 \mid x_t = x, c]. \tag{32}$$

Combining equation 31 and equation 32 and eliminating $\mathbb{E}[x_1 \mid x_t = x, c]$ gives

$$\mathbb{E}[x_0 \mid x_t = x, c] = x - t\, v^*(x, t, c). \tag{33}$$

Next, rewrite equation 28 as a Gaussian corruption of x_0. Let $y := \frac{x_t}{1-t}$ (for $t < 1$). Then

$$y = x_0 + \frac{t}{1-t}\, x_1 = x_0 + \sigma_t\, \varepsilon, \qquad \sigma_t := \frac{t}{1-t}, \;\; \varepsilon \sim \mathcal{N}(0, I). \tag{34}$$

Denote the marginal density of y by $p_Y(y \mid c)$. A standard Tweedie/Stein identity for Gaussian observations implies

$$\mathbb{E}[x_0 \mid y, c] = y + \sigma_t^2 \nabla_y \log p_Y(y \mid c). \tag{35}$$

Since $y = x/(1-t)$ is an invertible scaling, the densities satisfy $p_t(x \mid c) = (1-t)^{-d} p_Y(x/(1-t) \mid c)$, and thus

$$\nabla_x \log p_t(x \mid c) = \frac{1}{1-t}\, \nabla_y \log p_Y(y \mid c)\Big|_{y = x/(1-t)}. \tag{36}$$

Substituting equation 36 into equation 35 and evaluating at $y = x/(1-t)$ gives

$$\mathbb{E}[x_0 \mid x_t = x, c] = \frac{x}{1-t} + \frac{t^2}{(1-t)^2}\,(1-t)\,\nabla_x \log p_t(x \mid c) = \frac{x}{1-t} + \frac{t^2}{1-t}\,\nabla_x \log p_t(x \mid c). \tag{37}$$

Equating equation 37 with equation 33 yields

$$x - tv^*(x, t, c) = \frac{x}{1-t} + \frac{t^2}{1-t}\,\nabla_x \log p_t(x \mid c),$$

which rearranges to equation 30.

A practical score estimator using v_θ. Replacing v^* by v_θ in equation 30 yields an explicit score estimator requiring only the velocity network:

$$\widehat{\nabla_x \log p_t(x \mid c)} = -\frac{1}{t}\, x - \frac{1-t}{t}\, v_\theta(x, t, c). \tag{38}$$

Marginal-preserving reverse SDE without an explicit score network. Recall the marginal-preserving reverse-time SDE derived in equation 26:

$$\mathrm{d}x(t) = \left(v_\theta(x(t), t, c) - \frac{\sigma(t)^2}{2}\, \nabla_x \log p_t(x(t) \mid c) \right) \mathrm{d}t + \sigma(t)\, \mathrm{d}\bar{W}(t), \qquad t : 1 \to 0. \tag{39}$$

Substituting the estimator equation 38 gives a closed-form drift depending only on (x, t, v_θ):

$$\mathrm{d}x(t) = \underbrace{\left(v_\theta(x(t),t,c) + \frac{\sigma(t)^2}{2t}\Big(x(t) + (1-t)\, v_\theta(x(t),t,c)\Big)\right)}_{=:\, b_\theta(x(t),t,c)} \mathrm{d}t + \sigma(t)\, \mathrm{d}\bar{W}(t), \qquad t : 1 \to 0. \tag{40}$$

Euler–Maruyama discretization and per-step Gaussian density. Introduce a grid $1 = t_K > t_{K-1} > \cdots > t_0 = 0$. Denote the discrete state at time t_τ by $x^{(\tau)}$, where $\tau = K, K-1, \ldots, 0$ is the discrete denoising step index (superscript). Discretizing equation 40 on this grid with step $\Delta t_\tau := t_{\tau-1} - t_\tau < 0$ yields

$$x^{(\tau-1)} = x^{(\tau)} + \Delta t_\tau\, b_\theta(x^{(\tau)}, t_\tau, c) + \sigma(t_\tau)\sqrt{|\Delta t_\tau|}\, \varepsilon_\tau, \qquad \varepsilon_\tau \sim \mathcal{N}(0, I). \tag{41}$$

Thus the one-step transition is Gaussian:

$$p_\theta(x^{(\tau-1)} \mid x^{(\tau)}, c) = \mathcal{N}\Big(x^{(\tau)} + \Delta t_\tau\, b_\theta(x^{(\tau)}, t_\tau, c),\ \ \sigma(t_\tau)^2 |\Delta t_\tau|\, I\Big). \tag{42}$$

This provides a tractable stepwise log-density for GRPO:

$$\begin{aligned} \log p_\theta(x^{(\tau-1)} \mid x^{(\tau)}, c) = &-\frac{d}{2}\log\big(2\pi\sigma(t_\tau)^2|\Delta t_\tau|\big) \\ &- \frac{1}{2\sigma(t_\tau)^2|\Delta t_\tau|}\Big\| x^{(\tau-1)} - x^{(\tau)} - \Delta t_\tau\, b_\theta(x^{(\tau)}, t_\tau, c)\Big\|_2^2. \end{aligned} \tag{43}$$

Therefore, $\log \pi_\theta(\tau \mid c)$ factorizes as a sum of per-step Gaussian log-densities, enabling direct computation of the GRPO ratio $r_i(\theta)$.

Numerical note. Equations equation 38 and equation 40 contain $1/t$. In practice, we avoid numerical instability by truncating the time grid away from zero, e.g., $t_0 = \tau_{\min} > 0$, or by annealing $\sigma(t) \to 0$ as $t \to 0$ so that the correction term vanishes near the endpoint.

E GRPO Loss on SDE Sampler

Gaussian transition and trajectory factorization. Because ε_τ is Gaussian, equation 17 induces a non-degenerate Gaussian transition kernel:

$$p_\theta(x^{(\tau-1)} \mid x^{(\tau)}, c) = \mathcal{N}\Big(x^{(\tau)} + \Delta t_\tau b_\theta(x^{(\tau)}, t_\tau, c), \sigma(t_\tau)^2|\Delta t_\tau| I\Big). \tag{44}$$

Let d denote the data dimension. The corresponding log-density is

$$\begin{aligned} \log p_\theta(x^{(\tau-1)} \mid x^{(\tau)}, c) = &-\frac{d}{2}\log\big(2\pi\sigma(t_\tau)^2|\Delta t_\tau|\big) \\ &- \frac{1}{2\sigma(t_\tau)^2|\Delta t_\tau|}\big\| x^{(\tau-1)} - x^{(\tau)} - \Delta t_\tau b_\theta(x^{(\tau)}, t_\tau, c)\big\|_2^2. \end{aligned} \tag{45}$$

Define a trajectory $\tau = \{x^{(K)}, x^{(K-1)}, \ldots, x^{(0)}\}$ with initial $x^{(K)} \sim p_{\text{noise}}$. Under the Markov property implied by equation 44, the trajectory density factorizes:

$$\begin{aligned} \pi_\theta(\tau \mid c) &= p_{\text{noise}}(x^{(K)}) \prod_{\tau=1}^{K} p_\theta(x^{(\tau-1)} \mid x^{(\tau)}, c), \\ \log \pi_\theta(\tau \mid c) &= \log p_{\text{noise}}(x^{(K)}) + \sum_{\tau=1}^{K} \log p_\theta(x^{(\tau-1)} \mid x^{(\tau)}, c). \end{aligned} \tag{46}$$

Trajectory likelihood ratio and step-wise closed form. In GRPO, trajectories are sampled from the behavior policy $\pi_{\theta_{\text{old}}}$. For any realized trajectory τ_i, the likelihood ratio is therefore computable as

$$r_i(\theta) = \frac{\pi_\theta(\tau_i \mid c)}{\pi_{\theta_{\text{old}}}(\tau_i \mid c)} = \exp\Big(\sum_{\tau=1}^{K}\big[\log p_\theta(x^{(i,\tau-1)} \mid x^{(i,\tau)}, c) - \log p_{\theta_{\text{old}}}(x^{(i,\tau-1)} \mid x^{(i,\tau)}, c)\big]\Big), \tag{47}$$

where the (fixed) initial term $\log p_{\text{noise}}(x^{(K)})$ cancels if the same noise prior is used for both policies.

Using equation 45 and assuming both policies share the same s_τ schedule, the constant terms cancel and each step-wise log-ratio admits the residual-difference form

$$\begin{aligned}\Delta_\tau^{(i)}(\theta) &\triangleq \log p_\theta(x^{(i,\tau-1)} \mid x^{(i,\tau)}, c) - \log p_{\theta_{\text{old}}}(x^{(i,\tau-1)} \mid x^{(i,\tau)}, c) \\ &= \frac{1}{2s_\tau^2}\Big(\big\|x^{(i,\tau-1)} - x^{(i,\tau)} - m_{\theta_{\text{old}}}(x^{(i,\tau)}, t_\tau, c)\big\|_2^2 \\ &\qquad - \big\|x^{(i,\tau-1)} - x^{(i,\tau)} - m_\theta(x^{(i,\tau)}, t_\tau, c)\big\|_2^2\Big). \end{aligned} \tag{48}$$

Consequently,

$$\log r_i(\theta) = \sum_{\tau=1}^{K} \Delta_\tau^{(i)}(\theta), \qquad r_i(\theta) = \exp\Big(\sum_{\tau=1}^{K} \Delta_\tau^{(i)}(\theta)\Big). \tag{49}$$

Equivalent noise-explicit form (implementation-friendly). Because τ_i is sampled from $\pi_{\theta_{\text{old}}}$, the realized transition satisfies

$$x^{(i,\tau-1)} = x^{(i,\tau)} + m_{\theta_{\text{old}}}(x^{(i,\tau)}, t_\tau, c) + s_\tau\, \varepsilon_\tau^{(i)}, \qquad \varepsilon_\tau^{(i)} \sim \mathcal{N}(0, I). \tag{50}$$

Define the drift difference

$$\delta m_\theta(x^{(\tau)}, t_\tau, c) \triangleq m_\theta(x^{(\tau)}, t_\tau, c) - m_{\theta_{\text{old}}}(x^{(\tau)}, t_\tau, c). \tag{51}$$

Then the residual under θ becomes $x^{(i,\tau-1)} - x^{(i,\tau)} - m_\theta(x^{(i,\tau)}, t_\tau, c) = s_\tau \varepsilon_\tau^{(i)} - \delta m_\theta(x^{(i,\tau)}, t_\tau, c)$. Substituting into equation 48 yields the closed form

$$\Delta_\tau^{(i)}(\theta) = \frac{1}{s_\tau}\Big\langle \varepsilon_\tau^{(i)},\, \delta m_\theta(x^{(i,\tau)}, t_\tau, c)\Big\rangle - \frac{1}{2s_\tau^2}\Big\|\delta m_\theta(x^{(i,\tau)}, t_\tau, c)\Big\|_2^2, \tag{52}$$

and therefore

$$\log r_i(\theta) = \sum_{\tau=1}^{K}\left[\frac{1}{s_\tau}\Big\langle \varepsilon_\tau^{(i)},\, \delta m_\theta(x^{(i,\tau)}, t_\tau, c)\Big\rangle - \frac{1}{2s_\tau^2}\Big\|\delta m_\theta(x^{(i,\tau)}, t_\tau, c)\Big\|_2^2\right]. \tag{53}$$

Group-relative advantage (GRPO). For a given condition/prompt c, GRPO samples a group of G trajectories $\{\tau_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot \mid c)$ and obtains scalar rewards $R(\tau_i, c)$. Define the group mean and (stabilized) standard deviation

$$\bar{R}_c \triangleq \frac{1}{G}\sum_{i=1}^{G} R(\tau_i, c), \qquad \sigma_c \triangleq \sqrt{\frac{1}{G}\sum_{i=1}^{G}\big(R(\tau_i, c) - \bar{R}_c\big)^2 + \epsilon}, \tag{54}$$

and the normalized group-relative advantage

$$A_i \triangleq \frac{R(\tau_i, c) - \bar{R}_c}{\sigma_c}. \tag{55}$$

GRPO clipped surrogate objective (trajectory-level PPO). Let $\epsilon \in (0, 1)$ denote the clipping parameter. The GRPO clipped surrogate loss is

$$L_{\text{clip}}(\theta) = -\mathbb{E}_c\left[\frac{1}{G}\sum_{i=1}^{G}\min\Big(r_i(\theta)\, A_i,\ \text{clip}(r_i(\theta), 1-\epsilon, 1+\epsilon)\, A_i\Big)\right], \tag{56}$$

where $r_i(\theta)$ is computed by equation 47–equation 49 (or equivalently equation 53).

Optional KL regularization to a reference policy. In many GRPO/PPO-style instantiations, a KL penalty to a fixed reference policy π_{ref} is added. When both transitions share the same covariance $s_\tau^2 I$, the per-step KL has the closed form

$$\text{KL}\Big(p_\theta(\cdot \mid x^{(\tau)}, c) \,\|\, p_{\text{ref}}(\cdot \mid x^{(\tau)}, c)\Big) = \frac{1}{2s_\tau^2}\Big\|m_\theta(x^{(\tau)}, t_\tau, c) - m_{\text{ref}}(x^{(\tau)}, t_\tau, c)\Big\|_2^2. \tag{57}$$

A Monte-Carlo trajectory-level penalty is then

$$L_{\mathrm{KL}}(\theta) = \beta\, \mathbb{E}_c \left[\frac{1}{G} \sum_{i=1}^{G} \sum_{\tau=1}^{K} \frac{1}{2s_\tau^2} \left\| m_\theta(x^{(i,\tau)}, t_\tau, c) - m_{\mathrm{ref}}(x^{(i,\tau)}, t_\tau, c) \right\|_2^2 \right], \tag{58}$$

with coefficient $\beta \geq 0$.

Final GRPO training objective. Combining equation 56 and equation 58 yields the overall objective

$$L_{\mathrm{GRPO}}(\theta) = L_{\mathrm{clip}}(\theta) + L_{\mathrm{KL}}(\theta). \tag{59}$$

F RL Fine-Tuning Induces Responsibility-Matching Router Updates

Goal. We restate and prove Theorem 3.1 from the main text, showing that under policy-gradient RL fine-tuning, the router logits move in the advantage-signed direction of matching the posterior responsibility.

Theorem F.1 (Router-logit update under policy-gradient RL)**.** *At a routing layer, the router produces an expert prior $g_\phi(e \mid x^{(\tau)})$ over $e \in \{1, \ldots, E\}$ given router input $x^{(\tau)}$, where τ is the discrete denoising step index (superscript), and $z_e = z_\phi(x^{(\tau)})_e$ is the corresponding router logit. Let $p_\theta(x^{(\tau-1)} \mid x^{(\tau)}, e)$ be the expert-conditional likelihood. Define the posterior responsibility*

$$q_\tau(e \mid x^{(\tau)}, x^{(\tau-1)}) \triangleq \frac{g_\phi(e \mid x^{(\tau)})\, p_\theta(x^{(\tau-1)} \mid x^{(\tau)}, e)}{\sum_{j=1}^{E} g_\phi(j \mid x^{(\tau)})\, p_\theta(x^{(\tau-1)} \mid x^{(\tau)}, j)}. \tag{60}$$

Then under policy-gradient RL fine-tuning, the router-logit update satisfies

$$\Delta z_e \;\propto\; A \left(q_\tau(e \mid x^{(\tau)}, x^{(\tau-1)}) - g_\phi(e \mid x^{(\tau)}) \right), \tag{61}$$

where A is the advantage associated with the realized result $x^{(\tau-1)}$.

F.1 Setup: a single MoE routing site

Consider a single MoE site with router input $x^{(\tau)}$ and observed outcome o (e.g., a token, or a diffusion one-step transition $x^{(\tau-1)}$). The router defines logits $\{z_e(x^{(\tau)}; \phi)\}_{e=1}^{E}$ and the categorical prior

$$g_\phi(e \mid x^{(\tau)}) \triangleq g_e = \frac{\exp(z_e)}{\sum_{j=1}^{E} \exp(z_j)}. \tag{62}$$

Each expert e defines an expert-conditional likelihood $p_{\psi_e}(o \mid x^{(\tau)}, e)$; denote $p_e \triangleq p_{\psi_e}(o \mid x^{(\tau)}, e)$. The induced mixture likelihood is

$$p_\theta(o \mid x^{(\tau)}) = \sum_{j=1}^{E} g_j\, p_j, \qquad \theta = (\phi, \{\psi_e\}_{e=1}^{E}). \tag{63}$$

Given $(x^{(\tau)}, o)$, the posterior responsibility is

$$q_e(x^{(\tau)}, o) \triangleq \Pr(e \mid x^{(\tau)}, o) = \frac{g_e p_e}{\sum_{j=1}^{E} g_j p_j}, \qquad \sum_{e=1}^{E} q_e(x^{(\tau)}, o) = 1. \tag{64}$$

F.2 Key identity: router gradient equals responsibility minus prior

Let $S \triangleq \sum_{j=1}^{E} g_j p_j$ so that $\log p_\theta(o \mid x^{(\tau)}) = \log S$. Since p_j does not depend on router logits, we have

$$\frac{\partial}{\partial z_e} \log p_\theta(o \mid x^{(\tau)}) = \frac{1}{S} \sum_{j=1}^{E} \frac{\partial g_j}{\partial z_e}\, p_j. \tag{65}$$

Using the softmax derivative $\frac{\partial g_j}{\partial z_e} = g_j(\mathbf{1}[j=e] - g_e)$,

$$\frac{\partial}{\partial z_e}\log p_\theta(o \mid x^{(\tau)}) = \frac{1}{S}\sum_{j=1}^{E} g_j(\mathbf{1}[j=e] - g_e)p_j$$
$$= \frac{1}{S}\big(g_e p_e - g_e \sum_{j=1}^{E} g_j p_j\big) = \frac{g_e p_e}{S} - g_e = q_e(x^{(\tau)}, o) - g_e. \tag{66}$$

Therefore,

$$\boxed{\frac{\partial}{\partial z_e}\log p_\theta(o \mid x^{(\tau)}) = q_e(x^{(\tau)}, o) - g_\phi(e \mid x^{(\tau)}).} \tag{67}$$

F.3 Proof of Theorem F.1

Policy-gradient RL fine-tuning (e.g., REINFORCE/PPO core term) yields the sample-wise update direction

$$\Delta\theta \;\propto\; A\,\nabla_\theta \log p_\theta(o \mid x^{(\tau)}). \tag{68}$$

Restricting to router logits z_e and applying equation 67,

$$\Delta z_e \;\propto\; A\,\frac{\partial}{\partial z_e}\log p_\theta(o \mid x^{(\tau)}) = A\,(q_e(x^{(\tau)}, o) - g_e), \tag{69}$$

which is exactly equation 61. □

Remark (sign asymmetry). When $A > 0$, the update increases logits for experts with $q_e > g_e$ (experts more supported by the realized outcome) and decreases logits for experts with $q_e < g_e$; when $A < 0$ the direction flips, discouraging experts most responsible for low-reward outcomes.

Remark (Differentiable Relaxation and Top-1 Implementation). Theorem 3.1 uses a differentiable soft-router relaxation, while the implemented layer uses weighted hard Top-1 routing. If $e^* = \arg\max_e g_e$, the forward pass activates only expert e^* and multiplies its output by g_{e^*}. The gate probability therefore remains differentiable, with

$$\frac{\partial g_{e^*}}{\partial z_e} = g_{e^*}\big(\mathbf{1}[e=e^*] - g_e\big). \tag{70}$$

This derivative preserves the selected-expert-versus-router-prior form of the gate contribution, but the complete hard Top-1 forward is not the same probability model as the soft mixture. We therefore use the soft responsibility q_e as a relaxation-based qualitative description, while Figure 3(a,b) verifies the predicted sign and noise trends using the actual weighted hard Top-1 implementation.

G Diffusion Noise Controls Responsibility–Prior Deviation

Goal. We restate and prove the high-noise contraction result, which quantifies how diffusion noise suppresses the deviation between posterior responsibilities and router priors. The result is an upper-bound expansion for large noise; it is not a lower bound for the low-noise regime.

Theorem G.1 (High-noise responsibility contraction)**.** *When the variance scale ν_τ at SDE step τ is large, where τ denotes the discrete denoising step index (superscript), the responsibility–prior deviation satisfies*

$$q_\tau(e \mid x^{(\tau)}, x^{(\tau-1)}) - g_\tau(e \mid x^{(\tau)}) = O\Big(\frac{1}{\nu_\tau}\Big). \tag{71}$$

G.1 Setup: one diffusion step as an MoE mixture

Consider an SDE sampler step that transitions from $x^{(\tau)}$ to $x^{(\tau-1)}$ (conditioning omitted for brevity). At an MoE site inside the denoiser, the one-step kernel can be written as

$$p_\theta(x^{(\tau-1)} \mid x^{(\tau)}) = \sum_{e=1}^{E} g_{\tau,e}\, p_{\tau,e}, \qquad g_{\tau,e} \triangleq g_\phi(e \mid x^{(\tau)}), \quad p_{\tau,e} \triangleq p_{\psi_e}(x^{(\tau-1)} \mid x^{(\tau)}, e). \tag{72}$$

The posterior responsibility is

$$q_{\tau,e} \triangleq \Pr(e \mid x^{(\tau)}, x^{(\tau-1)}) = \frac{g_{\tau,e} p_{\tau,e}}{\sum_{j=1}^{E} g_{\tau,j} p_{\tau,j}}, \qquad \sum_{e=1}^{E} q_{\tau,e} = 1. \tag{73}$$

G.2 Responsibilities as a temperature-softmax of energies

For each expert e, define an energy (up to expert-independent constants)

$$E_{\tau,e} \triangleq -\log p_{\tau,e} + C_\tau, \tag{74}$$

so that $p_{\tau,e} \propto \exp(-E_{\tau,e})$. Substituting into equation 73 yields

$$q_\tau = \mathrm{softmax}\big(\log g_\tau - E_\tau\big). \tag{75}$$

Gaussian local form and variance scale. A standard discretization yields (approximately) Gaussian one-step conditionals with expert-dependent mean and (typically) expert-shared covariance:

$$p_{\tau,e} = \mathcal{N}\big(x^{(\tau-1)};\, \mu_{\tau,e},\, \Sigma_\tau\big). \tag{76}$$

Dropping expert-independent terms in the negative log-likelihood gives the residual energy

$$E_{\tau,e} = \frac{1}{2}\big(x^{(\tau-1)} - \mu_{\tau,e}\big)^\top \Sigma_\tau^{-1} \big(x^{(\tau-1)} - \mu_{\tau,e}\big). \tag{77}$$

In the isotropic case $\Sigma_\tau = s_\tau^2 I$, define the variance scale $\nu_\tau \triangleq 2s_\tau^2$ and the unscaled residual $\widetilde{E}_{\tau,e} \triangleq \|x^{(\tau-1)} - \mu_{\tau,e}\|_2^2$, so that $E_{\tau,e} = \widetilde{E}_{\tau,e}/\nu_\tau$ and hence

$$\boxed{q_\tau = \mathrm{softmax}\Big(\log g_\tau - \frac{1}{\nu_\tau}\widetilde{E}_\tau\Big).} \tag{78}$$

G.3 Proof of Theorem G.1 (high-noise expansion)

Let $u \triangleq \log g_\tau$ and $\delta_\tau \triangleq -\widetilde{E}_\tau/\nu_\tau$ so that equation 78 becomes $q_\tau = \mathrm{softmax}(u + \delta_\tau)$. Since $\mathrm{softmax}(u) = g_\tau$, a first-order Taylor expansion around u gives

$$q_{\tau,e} = g_{\tau,e} + \sum_{j=1}^{E} \frac{\partial\, \mathrm{softmax}_e(v)}{\partial v_j}\bigg|_{v=u} \delta_{\tau,j} + O(\|\delta_\tau\|^2). \tag{79}$$

Using the softmax Jacobian $\frac{\partial S_e}{\partial v_j} = S_e(\mathbf{1}[e=j] - S_j)$ and evaluating at $S(u) = g_\tau$, we obtain

$$q_{\tau,e} - g_{\tau,e} = g_{\tau,e}\Big(\delta_{\tau,e} - \sum_{j=1}^{E} g_{\tau,j}\delta_{\tau,j}\Big) + O(\|\delta_\tau\|^2). \tag{80}$$

Substituting $\delta_{\tau,e} = -\widetilde{E}_{\tau,e}/\nu_\tau$ yields the explicit scaling

$$q_{\tau,e} - g_{\tau,e} = -\frac{1}{\nu_\tau}\, g_{\tau,e}\Big(\widetilde{E}_{\tau,e} - \mathbb{E}_{j\sim g_\tau}[\widetilde{E}_{\tau,j}]\Big) + O\Big(\frac{1}{\nu_\tau^2}\Big). \tag{81}$$

Therefore, as $\nu_\tau \to \infty$, $q_{\tau,e} - g_{\tau,e} = O(1/\nu_\tau)$, which proves equation 71. □

Remark (implication for routing updates along denoising). Under SDE sampling, ν_τ generally decreases over denoising. Equation equation 81 shows that the high-noise expansion becomes less informative as ν_τ decreases; because q_τ and g_τ are probability distributions, it does not by itself imply an unbounded low-noise deviation. In our configurations, we empirically observe larger late-step routing changes, which motivates the noise-scaled regularization used by ANA-LB.

H SEQUENCE-LEVEL IMPORTANCE WEIGHTING FOR DIFFUSION-RL

Goal. We characterize the local T-dependent variance difference between the geometric-mean surrogate ratio and the standard trajectory ratio on diffusion chains. The statements in this section use a fixed-state/local approximation: $\mathcal{C}_i$ denotes conditioning on the prompt, the denoising states, and the local drift differences, which are held fixed while the step noises are analyzed. It is not a conditioning on the complete random trajectory in the general adaptive setting.

Theorem H.1 (Local geometric-mean surrogate and T-scaling)**.** *Let $\tau_i = \{x_i^{(T)}, \dots, x_i^{(0)}\}$ denote the states of a T-step diffusion trajectory, and interpret the following statement under the fixed-state conditioning $\mathcal{C}_i$ defined above; the step noises are conditionally independent under this local approximation. Here τ denotes the discrete denoising step index (superscript) and i is the sample index (subscript). Define the step-wise likelihood ratio and its log-increment*

$$w_{i,\tau}(\theta) \triangleq \frac{p_\theta(x_i^{(\tau-1)} \mid x_i^{(\tau)}, c)}{p_{\theta_{old}}(x_i^{(\tau-1)} \mid x_i^{(\tau)}, c)} \quad (>0), \qquad \Delta_{i,\tau}(\theta) \triangleq \log w_{i,\tau}(\theta), \tag{82}$$

the standard trajectory ratio, which factorizes as the product of the step-wise ratios, $r_i(\theta) \triangleq \prod_{\tau=1}^{T} w_{i,\tau}(\theta)$, and the geometric-mean surrogate ratio

$$\bar{r}_i(\theta) \triangleq \big(r_i(\theta)\big)^{1/T} = \exp\Big(\frac{1}{T}\sum_{\tau=1}^{T} \Delta_{i,\tau}(\theta)\Big). \tag{83}$$

Assume both π_θ and $\pi_{\theta_{old}}$ share the same $p_{noise}(x_i^{(T)})$ and the same variance schedule $\{s_\tau\}_{\tau=1}^{T}$, and each transition is Gaussian:

$$p_\theta(x_i^{(\tau-1)} \mid x_i^{(\tau)}, c) = \mathcal{N}\Big(x_i^{(\tau)} + m_\theta(x_i^{(\tau)}, t_\tau, c),\ s_\tau^2 I\Big). \tag{84}$$

Let $\delta m_\theta(x_i^{(\tau)}, t_\tau, c) \triangleq m_\theta(x_i^{(\tau)}, t_\tau, c) - m_{\theta_{old}}(x_i^{(\tau)}, t_\tau, c)$ and $a_{i,\tau}(\theta) \triangleq \|\delta m_\theta(x_i^{(\tau)}, t_\tau, c)\|_2 / s_\tau$. Then conditioned on $(x_i^{(\tau)}, c)$,

$$\Delta_{i,\tau}(\theta) = \frac{1}{s_\tau}\Big\langle \varepsilon_{i,\tau},\ \delta m_\theta(x_i^{(\tau)}, t_\tau, c)\Big\rangle - \frac{1}{2s_\tau^2}\Big\|\delta m_\theta(x_i^{(\tau)}, t_\tau, c)\Big\|_2^2, \tag{85}$$

where $\varepsilon_{i,\tau} \sim \mathcal{N}(0, I)$.

Under the same fixed-state approximation, each increment satisfies $\Delta_{i,\tau} \sim \mathcal{N}(-a_{i,\tau}^2/2, a_{i,\tau}^2)$. Consequently,

$$\mathbb{E}[\log \bar{r}_i \mid c, \mathcal{C}_i] = -\frac{1}{2T}\sum_{\tau=1}^{T} a_{i,\tau}^2, \qquad \mathbb{E}[\bar{r}_i \mid c, \mathcal{C}_i] = \exp\left(-\frac{T-1}{2T^2}\sum_{\tau=1}^{T} a_{i,\tau}^2\right) \le 1. \tag{86}$$

Thus, the geometric mean introduces a systematic local bias in exchange for reducing the variance of the accumulated log-ratio. These expectations are a local illustration and are not asserted as an exact unconditional characterization of an adaptive denoising chain.

Moreover, for any $\epsilon \in (0, 1)$, the centered log-ratio fluctuations satisfy the scalings

$$\begin{aligned} \mathbb{P}\big(\big|\log \bar{r}_i(\theta) - \mu_i^{\bar{r}}\big| \ge \epsilon \ \big|\ c, \mathcal{C}_i\big) &\le \exp\big(-\Omega(T\,\epsilon^2)\big), \\ \mathbb{P}\big(\big|\log r_i(\theta) - \mu_i^{r}\big| \ge \epsilon \ \big|\ c, \mathcal{C}_i\big) &\le \exp\big(-\Omega(\epsilon^2/T)\big), \end{aligned} \tag{87}$$

where $\mu_i^{\bar{r}}$ and μ_i^{r} are the corresponding conditional means and $\Omega(\cdot)$ hides constants depending on $\bar{a}_i(\theta)^2 \triangleq \frac{1}{T}\sum_{\tau=1}^{T} a_{i,\tau}(\theta)^2$.

H.1 STEP LOG-RATIO UNDER GAUSSIAN TRANSITIONS

Lemma H.2 (Closed form of $\Delta_{i,\tau}(\theta)$)**.** *Under equation 84 and sampling $\tau_i \sim \pi_{\theta_{old}}(\cdot \mid c)$, the behavior step can be written as*

$$x_i^{(\tau-1)} = x_i^{(\tau)} + m_{\theta_{old}}(x_i^{(\tau)}, t_\tau, c) + s_\tau\,\varepsilon_{i,\tau}, \qquad \varepsilon_{i,\tau} \sim \mathcal{N}(0, I). \tag{88}$$

Then conditioned on $(x_i^{(\tau)}, c)$, $\Delta_{i,\tau}(\theta)$ equals equation 85 and satisfies

$$\Delta_{i,\tau}(\theta) \ \big|\ (x_i^{(\tau)}, c) \ \sim\ \mathcal{N}\Big(-\tfrac{1}{2}a_{i,\tau}(\theta)^2,\ a_{i,\tau}(\theta)^2\Big). \tag{89}$$

Proof. Fix τ and abbreviate $x = x_i^{(\tau)}$ and $y = x_i^{(\tau-1)}$. Under equation 84,

$$\log p_\theta(y \mid x, c) = -\frac{d}{2}\log(2\pi s_\tau^2) - \frac{1}{2s_\tau^2}\big\|y - x - m_\theta(x, t_\tau, c)\big\|_2^2,$$

and the same formula holds for θ_{old} with $m_{\theta_{\text{old}}}$. Thus

$$\begin{aligned}
\Delta_{i,\tau}(\theta) &= \log \frac{p_\theta(y \mid x, c)}{p_{\theta_{\text{old}}}(y \mid x, c)} = -\frac{1}{2s_\tau^2}\Big(\|y - x - m_\theta\|_2^2 - \|y - x - m_{\theta_{\text{old}}}\|_2^2\Big) \\
&= -\frac{1}{2s_\tau^2}\Big(\|(y - x - m_{\theta_{\text{old}}}) - \delta m_\theta\|_2^2 - \|y - x - m_{\theta_{\text{old}}}\|_2^2\Big) \\
&= -\frac{1}{2s_\tau^2}\Big(\|y - x - m_{\theta_{\text{old}}}\|_2^2 - 2\langle y - x - m_{\theta_{\text{old}}}, \delta m_\theta\rangle + \|\delta m_\theta\|_2^2 - \|y - x - m_{\theta_{\text{old}}}\|_2^2\Big) \\
&= \frac{1}{s_\tau^2}\langle y - x - m_{\theta_{\text{old}}}, \delta m_\theta\rangle - \frac{1}{2s_\tau^2}\|\delta m_\theta\|_2^2. \qquad (90)
\end{aligned}$$

Under the behavior rollout equation 88, we have $y - x - m_{\theta_{\text{old}}} = s_\tau \varepsilon_{i,\tau}$, yielding equation 85. Conditioned on $(x_i^{(\tau)}, c)$, $\langle \varepsilon_{i,\tau}, \delta m_\theta\rangle \sim \mathcal{N}(0, \|\delta m_\theta\|_2^2)$, so equation 89 follows with variance $(\|\delta m_\theta\|_2/s_\tau)^2 = a_{i,\tau}(\theta)^2$ and mean shifted by $-\frac{1}{2}a_{i,\tau}(\theta)^2$. □

H.2 MGF BOUND (CONDITIONAL SUB-GAUSSIANITY)

Define the centered increment

$$Z_{i,\tau}(\theta) \triangleq \Delta_{i,\tau}(\theta) - \mathbb{E}[\Delta_{i,\tau}(\theta) \mid x_i^{(\tau)}, c]. \qquad (91)$$

Lemma H.3 (Conditional sub-Gaussian MGF)**.** *Conditioned on $(x_i^{(\tau)}, c)$, $Z_{i,\tau}(\theta)$ is sub-Gaussian with proxy variance $a_{i,\tau}(\theta)^2$, i.e., for all $\lambda \in \mathbb{R}$,*

$$\mathbb{E}\Big[\exp(\lambda Z_{i,\tau}(\theta)) \;\Big|\; x_i^{(\tau)}, c\Big] \le \exp\left(\frac{\lambda^2 a_{i,\tau}(\theta)^2}{2}\right). \qquad (92)$$

Proof. By Lemma H.2, conditioned on $(x_i^{(\tau)}, c)$,

$$\Delta_{i,\tau}(\theta) = -\tfrac{1}{2}a_{i,\tau}^2 + a_{i,\tau}\,\xi, \qquad \xi \sim \mathcal{N}(0,1). \qquad (93)$$

Hence $Z_{i,\tau}(\theta) = a_{i,\tau}\xi$ and

$$\mathbb{E}\Big[e^{\lambda Z_{i,\tau}} \mid x_i^{(\tau)}, c\Big] = \mathbb{E}[e^{\lambda a_{i,\tau}\xi}] = \exp\left(\frac{\lambda^2 a_{i,\tau}^2}{2}\right), \qquad (94)$$

which implies equation 92 (in fact equality holds). □

H.3 FULL CONCENTRATION PROOFS FOR $\log \bar{r}_i$ AND $\log r_i$

Assume that conditioned on $(c, \mathcal{C}_i)$, the noises $\{\varepsilon_{i,\tau}\}_{\tau=1}^T$ are independent, so that $\{\Delta_{i,\tau}(\theta)\}_{\tau=1}^T$ are conditionally independent as well. Let

$$\mu_i^{\bar{r}} \triangleq \mathbb{E}[\log \bar{r}_i(\theta) \mid c, \mathcal{C}_i], \qquad \mu_i^{r} \triangleq \mathbb{E}[\log r_i(\theta) \mid c, \mathcal{C}_i], \qquad \bar{a}_i(\theta)^2 \triangleq \frac{1}{T}\sum_{\tau=1}^T a_{i,\tau}(\theta)^2. \qquad (95)$$

Proposition H.4 (Tail bound for $\log \bar{r}_i$)**.** *For any $u > 0$,*

$$\mathbb{P}\big(\big|\log \bar{r}_i(\theta) - \mu_i^{\bar{r}}\big| \ge u \;\big|\; c, \mathcal{C}_i\big) \le 2\exp\left(-\frac{Tu^2}{2\,\bar{a}_i(\theta)^2}\right). \qquad (96)$$

Proof. Write

$$\log \bar{r}_i(\theta) - \mu_i^{\bar{r}} = \frac{1}{T}\sum_{\tau=1}^T \Big(\Delta_{i,\tau}(\theta) - \mathbb{E}[\Delta_{i,\tau}(\theta) \mid c, \mathcal{C}_i]\Big) = \frac{1}{T}\sum_{\tau=1}^T Z_{i,\tau}(\theta), \qquad (97)$$

where $Z_{i,\tau}$ are conditionally independent and satisfy the MGF bound in Lemma H.3. For any $\lambda > 0$, by Markov's inequality,

$$\begin{aligned}
\mathbb{P}\big(\log \bar{r}_i - \mu_i^{\bar{r}} \geq u \mid c, \mathcal{C}_i\big) &= \mathbb{P}\left(\exp\left(\lambda \frac{1}{T}\sum_{\tau=1}^{T} Z_{i,\tau}\right) \geq e^{\lambda u} \,\middle|\, c, \mathcal{C}_i\right) \\
&\leq e^{-\lambda u}\, \mathbb{E}\left[\exp\left(\lambda \frac{1}{T}\sum_{\tau=1}^{T} Z_{i,\tau}\right) \,\middle|\, c, \mathcal{C}_i\right] \\
&= e^{-\lambda u} \prod_{\tau=1}^{T} \mathbb{E}\left[\exp\left(\frac{\lambda}{T} Z_{i,\tau}\right) \,\middle|\, c, \mathcal{C}_i\right] \\
&\leq e^{-\lambda u} \prod_{\tau=1}^{T} \exp\left(\frac{(\lambda/T)^2 a_{i,\tau}^2}{2}\right) \\
&= \exp\left(-\lambda u + \frac{\lambda^2}{2T^2}\sum_{\tau=1}^{T} a_{i,\tau}^2\right) = \exp\left(-\lambda u + \frac{\lambda^2}{2}\cdot\frac{\bar{a}_i^2}{T}\right). \qquad (98)
\end{aligned}$$

Minimizing the RHS over $\lambda > 0$ gives $\lambda^\star = Tu/\bar{a}_i^2$, yielding

$$\mathbb{P}\big(\log \bar{r}_i - \mu_i^{\bar{r}} \geq u \mid c, \mathcal{C}_i\big) \leq \exp\left(-\frac{Tu^2}{2\bar{a}_i^2}\right). \qquad (99)$$

The lower tail is analogous by applying the same argument to $-(\log \bar{r}_i - \mu_i^{\bar{r}})$, hence equation 96 by a union bound. □

Proposition H.5 (Tail bound for $\log r_i$)**.** *For any $u > 0$,*

$$\mathbb{P}(|\log r_i(\theta) - \mu_i^r| \geq u \mid c, \mathcal{C}_i) \leq 2\exp\left(-\frac{u^2}{2T\,\bar{a}_i(\theta)^2}\right). \qquad (100)$$

Proof. Now $\log r_i(\theta) - \mu_i^r = \sum_{\tau=1}^{T} Z_{i,\tau}(\theta)$. Repeating the Chernoff bound argument without the $1/T$ factor, for any $\lambda > 0$,

$$\begin{aligned}
\mathbb{P}(\log r_i - \mu_i^r \geq u \mid c, \mathcal{C}_i) &\leq e^{-\lambda u} \prod_{\tau=1}^{T} \mathbb{E}\big[e^{\lambda Z_{i,\tau}} \mid c, \mathcal{C}_i\big] \\
&\leq e^{-\lambda u} \prod_{\tau=1}^{T} \exp\left(\frac{\lambda^2 a_{i,\tau}^2}{2}\right) \\
&= \exp\left(-\lambda u + \frac{\lambda^2}{2}\sum_{\tau=1}^{T} a_{i,\tau}^2\right) = \exp\left(-\lambda u + \frac{\lambda^2}{2}\cdot T\bar{a}_i^2\right). \qquad (101)
\end{aligned}$$

Minimizing over $\lambda > 0$ gives $\lambda^\star = u/(T\bar{a}_i^2)$ and

$$\mathbb{P}(\log r_i - \mu_i^r \geq u \mid c, \mathcal{C}_i) \leq \exp\left(-\frac{u^2}{2T\bar{a}_i^2}\right).$$

The lower tail is symmetric, and the two-sided bound follows by union bound, giving equation 100. □

H.4 From log-ratio tails to deviation-from-one probability

Fix $\epsilon \in (0, 1)$ and define $\ell_+ = \log(1+\epsilon)$ and $\ell_- = \log(1-\epsilon) < 0$.

Corollary H.6 (Deviation-from-one: geometric mean)**.** *Let $X = \log \bar{r}_i(\theta)$ and $\mu_i^{\bar{r}} = \mathbb{E}[X \mid c, \mathcal{C}_i]$. Then*

$$\mathbb{P}(|\bar{r}_i(\theta) - 1| \geq \epsilon \mid c, \mathcal{C}_i) \leq \exp\left(-\frac{T(\ell_+ - \mu_i^{\bar{r}})^2}{2\bar{a}_i^2}\right) + \exp\left(-\frac{T(\mu_i^{\bar{r}} - \ell_-)^2}{2\bar{a}_i^2}\right). \qquad (102)$$

In particular, if $\mu_i^{\bar{r}} \in [\ell_-, \ell_+]$ *and* $\Delta_i \triangleq \min\{\ell_+ - \mu_i^{\bar{r}},\ \mu_i^{\bar{r}} - \ell_-\}$*, then*

$$\mathbb{P}(|\bar{r}_i(\theta) - 1| \geq \epsilon \mid c, \mathcal{C}_i) \leq 2\exp\left(-\frac{T\Delta_i^2}{2\bar{a}_i^2}\right) = \exp\big(-\Omega(T\epsilon^2)\big). \tag{103}$$

Proof. The event $\{|\bar{r}_i - 1| \geq \epsilon\}$ equals $\{\bar{r}_i \geq 1 + \epsilon\} \cup \{\bar{r}_i \leq 1 - \epsilon\}$. Since $\bar{r}_i = \exp(X)$, these are $\{X \geq \ell_+\}$ and $\{X \leq \ell_-\}$. Thus

$$\mathbb{P}(|\bar{r}_i - 1| \geq \epsilon \mid c, \mathcal{C}_i) \leq \mathbb{P}(X - \mu_i^{\bar{r}} \geq \ell_+ - \mu_i^{\bar{r}} \mid c, \mathcal{C}_i) + \mathbb{P}(\mu_i^{\bar{r}} - X \geq \mu_i^{\bar{r}} - \ell_- \mid c, \mathcal{C}_i).$$

Apply Proposition H.4 to each one-sided tail to obtain equation 102. The simplified bound equation 103 follows when $\mu_i^{\bar{r}} \in [\ell_-, \ell_+]$. □

Corollary H.7 (Deviation-from-one: standard trajectory ratio)**.** *Let* $Y = \log r_i(\theta)$ *and* $\mu_i^r = \mathbb{E}[Y \mid c, \mathcal{C}_i]$*. Then*

$$\mathbb{P}(|r_i(\theta) - 1| \geq \epsilon \mid c, \mathcal{C}_i) \leq \exp\left(-\frac{(\ell_+ - \mu_i^r)^2}{2T\bar{a}_i^2}\right) + \exp\left(-\frac{(\mu_i^r - \ell_-)^2}{2T\bar{a}_i^2}\right). \tag{104}$$

In particular, if $\mu_i^r \in [\ell_-, \ell_+]$ *and* $\tilde{\Delta}_i \triangleq \min\{\ell_+ - \mu_i^r,\ \mu_i^r - \ell_-\}$*, then*

$$\mathbb{P}(|r_i(\theta) - 1| \geq \epsilon \mid c, \mathcal{C}_i) \leq 2\exp\left(-\frac{\tilde{\Delta}_i^2}{2T\bar{a}_i^2}\right) = \exp\big(-\Omega(\epsilon^2/T)\big). \tag{105}$$

Proof. Identical to Corollary H.6, but using Proposition H.5 for the log-sum tail. □

Conclusion. Combining Corollaries H.6 and H.7 completes the proof of Theorem H.1. □

I Training Details

In this section, we provide the experimental settings, hyperparameters, and infrastructure details used in our study. Our implementation is based on the `RLinf` framework, extended to support Sparse Mixture-of-Experts (MoE) modules and our proposed stabilization techniques (GMR and ANA-LB). All experiments were conducted on a cluster of 8× NVIDIA A100 (80GB) GPUs using Fully Sharded Data Parallel (FSDP).

I.1 Model Architecture and Initialization

We utilize both π_0 and $\pi_{0.5}$ as our backbone policies. Structurally, these models are composed of three key components: a SigLIP vision encoder for visual feature extraction, a Gemma LLM backbone for language processing and feature integration, and a transformer-based Diffusion Transformer (DiT) decoder for action generation via flow matching.

For the MoE adaptation, we specifically target the DiT decoder. We replace the standard Feed-Forward Networks (FFNs) in the DiT layers with our **Shared-MoE** layers to enhance the policy's capacity for multi-task behaviors. The router weights are initialized from a normal distribution $\mathcal{N}(0, 0.01)$ to ensure near-uniform routing probabilities at the beginning of RL fine-tuning.

I.2 Hyperparameters and Optimization

We employ our proposed **Advantage- and Noise-Aware Load Balancing** and **Geometric-Mean Ratio** for all RL fine-tuning experiments. Our method extends the Group Relative Policy Optimization (GRPO) framework by incorporating geometric mean aggregation for flow-based likelihoods. The specific hyperparameters for the optimization process, our algorithm, and our proposed MoE mechanisms are detailed in Table 5.

Multi-seed aggregation. All simulation results obtained from our RL runs in Tables 1–3 are arithmetic means over three independent random seeds.

Table 5: Hyperparameters for $\pi_{0.5}$ MoE-RL Fine-tuning.

Hyperparameter	Value
General Optimization	
Number of RL Seeds	3
Optimizer	AdamW
Learning Rate (Policy)	5.6×10^{-6}
Weight Decay	0.0
Warmup Steps	100
Learning Rate Scheduler	Constant
Gradient Clipping (Global Norm)	1.0
Global Batch Size	256
GMR Algorithm (Based on GRPO)	
Group Size (G)	16
Discount Factor (γ)	0.99
Clip Range (ϵ)	0.2
KL Coefficient (β)	0.04
Update Epochs	4
Advantage Estimation	Group Relative (Avg. Baseline)
Flow Matching & GMR (Ours)	
Denoising Steps (T_{train}/T_{infer})	10
Log-Probability Calculation	Geometric Mean (GMR)
Action Space	Continuous (Delta Pose + Gripper)
Sparse MoE & ANA-LB (Ours)	
Total Experts (E)	8
Activated Experts (M)	1
Router Loss Coefficient (λ_{lb})	0.01
Load Balancing Type	Advantage- and Noise-Aware (ANA-LB)
ANA-LB Weight Schedule	$\lambda_{\text{ANA}}(t) = \sigma\sqrt{\frac{t}{1-t}}$
Base Noise Level (σ)	0.7

I.3 Task-Specific Training Details

Libero Training Details We evaluate our method on the **LIBERO** benchmark, encompassing the LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long suites. The benchmark is built on `robosuite` and simulates a 7-DoF robotic arm performing various daily manipulation skills (e.g., stacking, opening drawers, spatial re-arrangement).

- **Observation and Prompting:** The agent receives RGB images from an off-screen third-person camera, resized to a resolution of 224×224. The task instruction is structured with the specific template: "*What action should the robot take to [task_description]?*".
- **Action Space:** The action space consists of 7-dimensional continuous control, including the end-effector's 3D position (x, y, z), 3D rotation (roll, pitch, yaw), and binary gripper state (open/close).
- **Reward Configuration:** We utilize sparse binary rewards $(0/1)$, which are granted solely upon the successful completion of the target task. This uniform reward scheme ensures a standardized evaluation of the RL fine-tuning process across both benchmarks.

CALVIN Training Details We conduct experiments on the **CALVIN** benchmark, focusing on the challenging Zero-Shot Multi-Environment setting (denoted as **ABC→D**). In this protocol, the agent is fine-tuned via RL on scenes A, B, and C—which feature distinct textures and varying layouts of static elements (e.g., doors, drawers, lights)—and is subsequently evaluated on the unseen scene D.

- **Observation Space:** The policy operates on RGB inputs from two viewpoints: a static third-person camera and a wrist-mounted gripper camera.
- **Action Space:** The 7-dimensional continuous action space consists of relative end-effector position control (x, y, z), relative rotation control (roll, pitch, yaw), and a binary gripper command (open/close).
- **Chain Evaluation and Rewards:** The primary metric is the success rate of sequentially completing chains of 5 natural language instructions. For RL fine-tuning, we utilize sparse

binary rewards $(0/1)$, which are granted solely upon the successful completion of each individual subtask within the chain.

RoboTwin 2.0 Training Details We follow the official RoboTwin 2.0 evaluator on the `aloha-agilex` bimanual platform and the complete set of 50 tasks.

- **Observation and Prompting:** Each task uses the official language instruction and RGB observations from a head-mounted D435 camera and two wrist-mounted D435 cameras at the native 320×240 resolution. The clean and randomized configurations use the same camera layout; the latter additionally randomizes backgrounds, tabletop clutter, lighting, and tabletop height according to the official task configuration.
- **Action Space:** The policy outputs the official 14-dimensional bimanual joint action with `action-type=joint`.
- **Policy and Optimization:** We fine-tune π_0 and $\pi_{0.5}$ from the same SFT initialization with group-relative GRPO and the ten-step SDE sampler. The dense baseline uses Flow-SDE, whereas Ours uses weighted hard Top-1 MoE routing with ANA-LB and GMR.
- **Reward Configuration:** Training uses the sparse 0–1 task-success reward specified by the benchmark protocol.
- **Evaluation:** We evaluate each of the 50 tasks separately under `demo_clean` (Easy) and `demo_randomized` (Hard), using the official task success check and task-specific episode horizon. We report the mean success rate over the 50 tasks for each configuration.

I.4 Ablation Study Training Details

To analyze the contribution of each component, we conducted a systematic ablation study corresponding to the results presented in Section 4.2. The specific training configurations for each entry are defined as follows:

- **MoE + GRPO (Baseline):** The baseline setup utilizing standard Group Relative Policy Optimization (GRPO) adapted for MoE. In this setting, the policy log-probability is calculated via the standard summation over diffusion steps ($\log \pi = \sum_{t=1}^{T} \log p_t$), and **no auxiliary load balancing loss** is applied ($\lambda_{lb} = 0$).
- **+ LB (Standard Load Balancing):** We apply the same KL-divergence auxiliary loss formulation as in our proposed method, but **without** the advantage masking and noise-aware weighting. Specifically, we minimize the objective $\sum_{i=1}^{G} \sum_{k=1}^{K} \mathrm{KL}(g_\phi \,\|\, U(E))$ uniformly across all samples and diffusion timesteps, forcing the router to balance expert usage regardless of the advantage sign or noise level.
- **+ ANA-LB (Advantage- and Noise-Aware LB):** Replaces the standard LB with our proposed **ANA-LB** mechanism. While keeping the log-probability calculation as the standard summation, this setting enables the negative advantage mask ($\mathbb{I}_{A<0}$) and applies the step-dependent weighting schedule $\lambda_{\mathrm{ANA}}(t)$ to the router's auxiliary loss.
- **+ GMR (Geometric-Mean Ratio):** Replaces the baseline aggregation of step-wise log-probabilities with our proposed **Geometric-Mean Ratio (GMR)** formulation ($\log \pi = \frac{1}{T} \sum_{t=1}^{T} \log p_t$). This experiment isolates the stability benefits of the GMR estimator without applying any router regularization (i.e., $\lambda_{lb} = 0$).

J Task Examples

In this section, we provide detailed descriptions of the evaluation tasks from the LIBERO-Long benchmark used in our experiments. We first outline the specific instructions and embodied manipulation challenges for six tasks. Visual comparisons of typical failure modes versus successful executions for each task are presented in Figures 4 through 9 at the end of this section.

J.1 Task 1: Cabinet Manipulation

- **Instruction:** "Put the black bowl in the bottom drawer of the cabinet and close it."

- **Challenge (Strict Geometric Preconditions):** This is a typical long-horizon dependency task. Even if the bowl is visually inside the drawer, failing to push it deeply enough (Deep Insertion) creates a physical blockage. As illustrated in the top row of Figure 4, the failure occurs because the bowl's shallow placement mechanically obstructs the drawer's path, causing a deadlock during the closing phase.

J.2 Task 2: Microwave Interaction

- **Instruction:** "Put the yellow mug in the microwave and close it."
- **Challenge (Manipulation in Confined Volumes):** The interior of the microwave is narrow with restricted entry margins. The end-effector is highly prone to collision with the microwave walls or the door frame during the insertion phase. The top row of Figure 5 demonstrates this failure mode, where the agent fails to align perfectly with the narrow entry, causing the gripper to collide with the frame.

J.3 Task 3: Basket Packing

- **Instruction:** "Put both the alphabet soup and the tomato sauce in the basket."
- **Challenge (Deep Container Insertion):** The basket features high vertical walls. The robot must execute a specific "U-shaped" trajectory—lifting the object high above the rim before lowering it vertically. This difficulty is highlighted in the top row of Figure 6, where the agent attempts a direct linear path instead of lifting sufficiently, causing the object to strike the basket's edge.

J.4 Task 4: Stove Activation

- **Instruction:** "Turn on the stove and put the moka pot on it."
- **Challenge (Precise Alignment to Functional Region):** The task imposes strict semantic requirements. The robot cannot simply place the pot on any surface of the stove; it must align precisely with the center of the heating element. In the top row of Figure 7, although the pot is placed on the stove, the task fails because the agent misses the specific functional heating region.

J.5 Task 5: Semantic Sorting

- **Instruction:** "Put the white mug on the left plate and put the yellow mug on the right plate."
- **Challenge (Placement on Constrained Surfaces):** Plates have limited target areas and often feature curved geometries or rims. Compared to placing objects on a flat table, placing them on plates requires higher vertical control precision. As shown in the top row of Figure 8, the failure arises when the mug is placed on the plate's curved rim rather than its center, resulting in an unstable tilt.

J.6 Task 6: Crowded Placement

- **Instruction:** "Put both moka pots on the stove."
- **Challenge (Physical Stability):** When placing the second pot, air turbulence or minor physical contact can easily cause the first pot to slide off the stove's edge. The top row of Figure 9 captures this stability failure, where the agent accidentally knocks over the previously placed pot while attempting to position the second one in the crowded space.

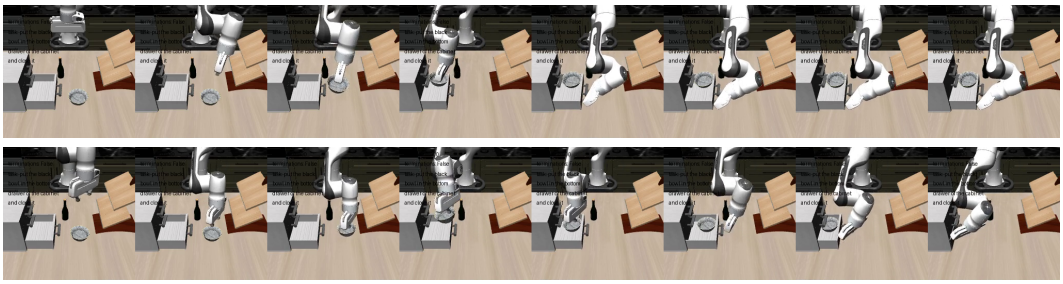


Figure 4: **Task 1 (Cabinet Manipulation):** Visual comparison (Top: Failure; Bottom: Success) showing how geometric precision (deep insertion) determines task success.

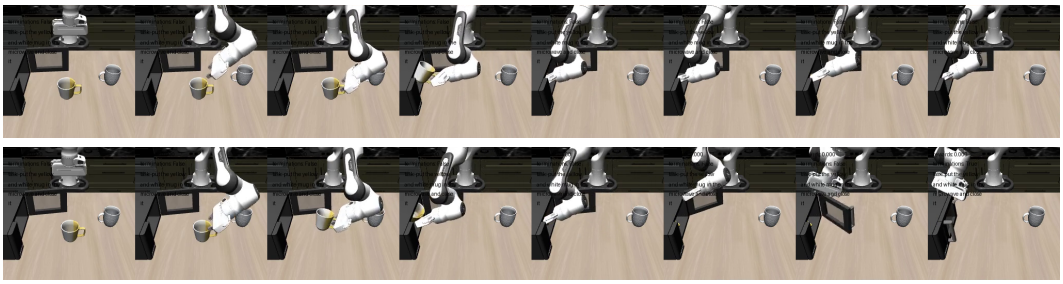


Figure 5: **Task 2 (Microwave Interaction):** Visual comparison (Top: Failure; Bottom: Success) demonstrating the challenge of operating within confined volumes under occlusion.



Figure 6: **Task 3 (Basket Packing):** Visual comparison (Top: Failure; Bottom: Success) showing the necessity of vertical clearance for deep container insertion.

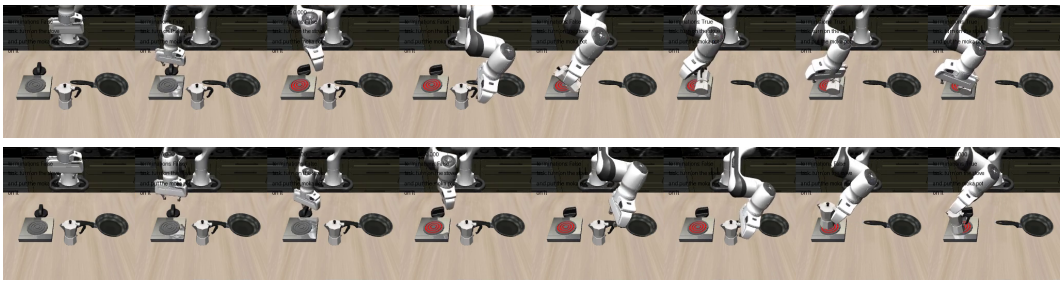


Figure 7: **Task 4 (Stove Activation):** Visual comparison (Top: Failure; Bottom: Success) highlighting the precision required to align objects with functional regions.

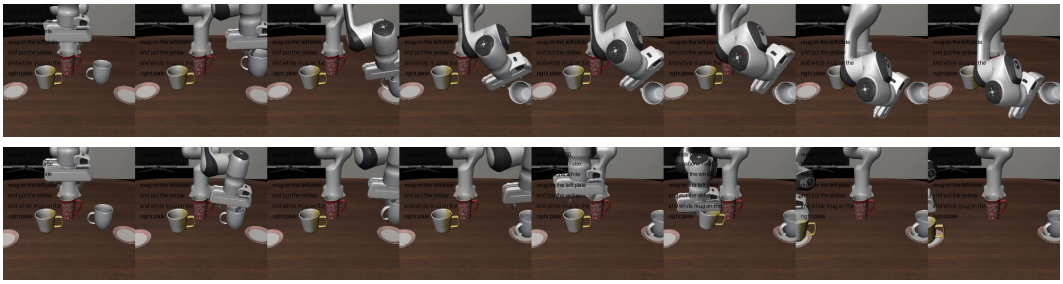


Figure 8: **Task 5 (Semantic Sorting):** Visual comparison (Top: Failure; Bottom: Success) illustrating the difficulty of stable placement on constrained, curved surfaces.

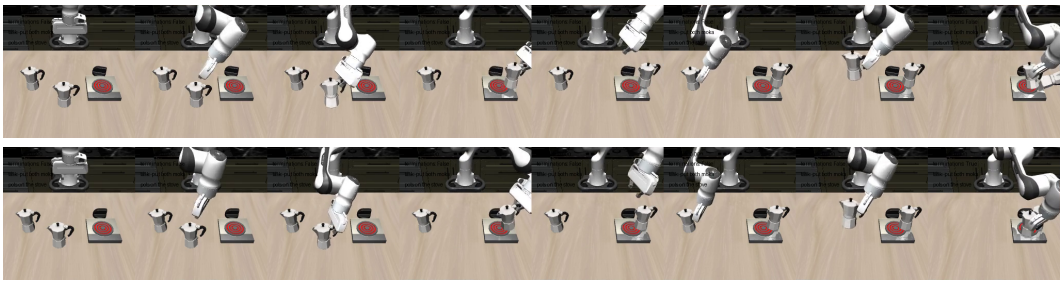


Figure 9: **Task 6 (Crowded Placement):** Visual comparison (Top: Failure; Bottom: Success) demonstrating the physical stability challenges in crowded spaces.

K Real-World Task Examples and Visual Rollouts

In this section, we provide a comprehensive overview of the real-world evaluation protocol deployed on the Piper robotic arm. We detail the 24 challenging manipulation tasks spanning six distinct capability dimensions. Subsequently, we provide representative frame-by-frame visual sequences for each category, demonstrating the stabilized **MoE-Flow Matching** policy's proficiency in handling extreme precision, long-horizon composition, and unstructured physical disturbances.

K.1 Detailed Task Evaluation Suite

To rigorously assess the closed-loop control and generalization of the policy, we designed 4 specific sub-tasks for each of the 6 capability dimensions. The complete task configurations are as follows:

- **1. Basic Relocation:** Validates fundamental grasping and spatial translation capabilities.
  1. Place the red bowl onto the blue plate.
  2. Place the blue plate into the cabinet.

3. Drop the white cap into the red bowl.
4. Move the pink block from the center of the workspace to a predefined point on the left edge.

- **2. Precision Alignment:** Requires extremely low control tolerance and high-frequency stabilization.
  1. Place the pink block precisely on top of the small white cap.
  2. Locate and grasp the transparent cup strictly by its narrow handle.
  3. Insert the red bowl into a constrained free area surrounded by obstacles.
  4. Align the pink block flush against the edge of the cabinet with zero protruding margin.
- **3. Long-Horizon Composition:** Tests temporal consistency across multi-stage, sequential dependencies.
  1. Place the red bowl and a transparent bowl onto the blue plate, then pick up the entire assembly and transfer it into the cabinet.
  2. Place the pink block into the transparent bowl, then move the combined objects onto top of the cabinet.
  3. Upright a fallen red bowl, then accurately cover it with the blue plate.
  4. Collect transparent bowls from both the cabinet top and the interior, then stack them onto the blue plate.
- **4. Dynamic Disturbance:** Evaluates real-world reactivity and closed-loop recovery under non-stationary environments.
  1. Manually shift the target red bowl while the arm is approaching to drop the white cap.
  2. Physically knock the pink block out of the gripper after a successful grasp.
  3. Disrupt the arm's trajectory during motion.
  4. Rotate the cup's handle during the approach phase.
- **5. Visual Occlusion:** Assesses perceptual robustness when visual information is incomplete.
  1. Retrieve the pink block from the red bowl.
  2. Locate and retrieve the white cap from the cabinet.
  3. Grasp the transparent cup that is partially covered by the blue plate's edge.
  4. Identify and extract the red bowl from a crowded scene with multiple surrounding distractors.
- **6. Material Generalization:** Tests the ability to overcome visual noise such as refraction and reflection.
  1. Grasp the transparent cup inside the blue plate.
  2. Stack one transparent bowl onto another.
  3. Retrieve a transparent bowl from the cabinet.
  4. Place the transparent bowl upside down over the transparent cup.

K.2 Representative Visual Rollouts

The following sequences illustrate the model's performance in one representative task from each of the six categories:

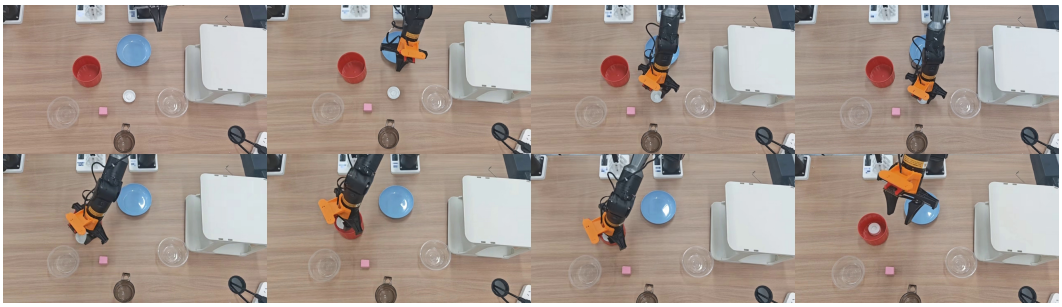


Figure 10: **Basic Relocation:** Placing a white cap into a red bowl. This sequence demonstrates that our MoE-Flow Matching framework preserves fundamental manipulation primitives while maintaining smooth trajectories during the approach and release phases.

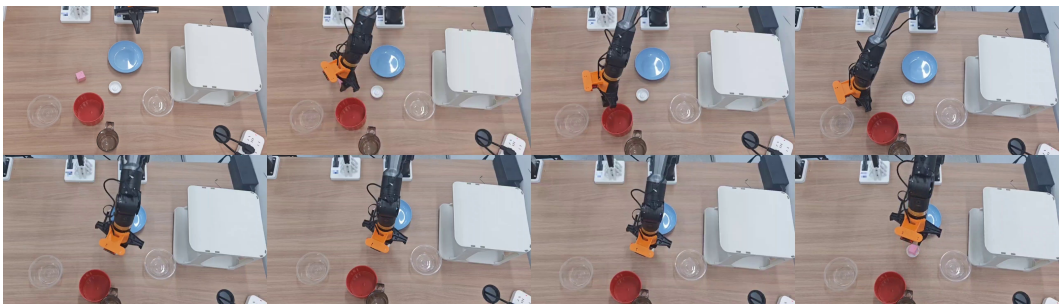


Figure 11: **Precision Alignment:** Stacking a pink block onto a white cap. The task demands sub-centimeter accuracy. The policy routes to specialized experts to suppress end-effector jitter, ensuring a stable and centered placement.



Figure 12: **Long-Horizon Composition:** Transferring a multi-object assembly into the cabinet. The model must maintain a consistent latent representation of the ”assembly” state across the pick, transport, and place stages without temporal drift.

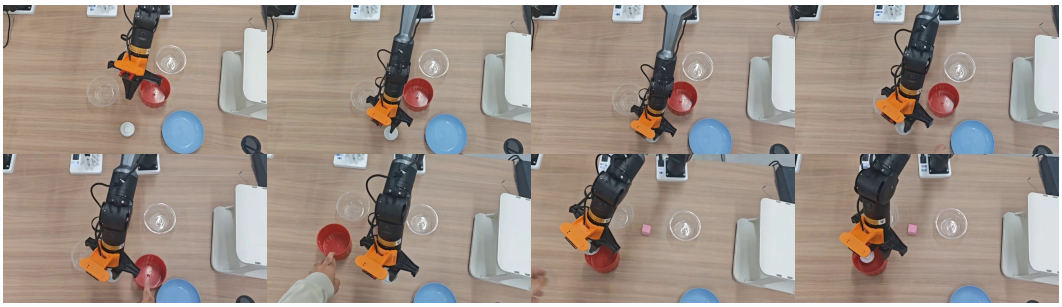


Figure 13: **Dynamic Disturbance:** Real-time recovery during target displacement. When the red bowl is moved, the policy leverages continuous flow-matching to dynamically warp its velocity field, successfully capturing the target despite the non-stationary goal.

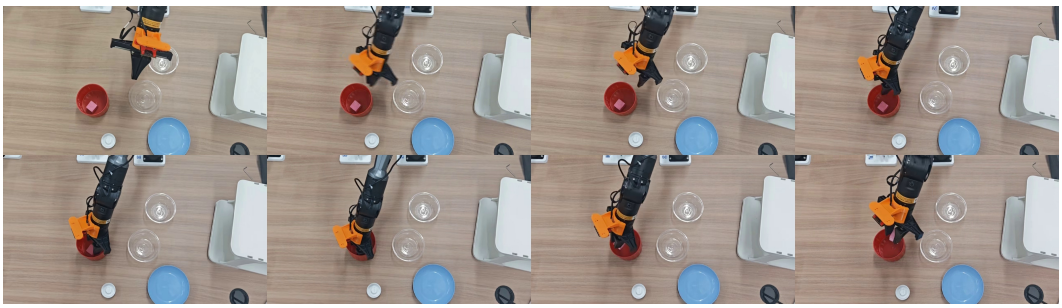


Figure 14: **Visual Occlusion:** Retrieving a block from a deep bowl. Despite the primary camera losing most of the target's visual features due to occlusion from the bowl's walls, the model robustly infers the object's centroid for a successful grasp.



Figure 15: **Material Generalization & Low-Light Retrieval:** Extracting a transparent bowl from the cabinet. This challenging case requires the MoE architecture to disentangle the weak visual boundaries of the transparent material from the dark cabinet background for precise localization.