DREAMTAMER: FROM LUCKY SAMPLES TO RELIABLE FUTURES IN ACTION CONDITIONED WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Action-conditioned world models (ACWMs) predict future observations from visual histories and robot actions, and must faithfully capture the consequences of those actions. However, even with identical observations and actions, pretrained ACWMs can produce substantially different interaction outcomes across random seeds, making faithful prediction inconsistent across samples. To understand this variability, we conduct denoising-stage branching experiments and find that early-stage stochasticity strongly affects prediction fidelity and task-critical interaction dynamics, whereas later-stage stochasticity primarily affects visual details. This finding motivates a reinforcement learning post-training framework that concentrates optimization on denoising transitions with greater influence on prediction fidelity. Specifically, we combine timestep-aware gradient correction with variance-aware timestep sampling to direct learning toward these influential transitions. To further improve reliability across samples, we introduce a downside penalty that targets low-fidelity predictions without directly penalizing favorable deviations. Experiments on Ctrl-World (Guo et al., 2026), DreamDojo (Gao et al., 2026), and Genie Envisioner (Liao et al., 2026) demonstrate improvements in average prediction quality, the quality of low-fidelity predictions, and consistency across repeated samples. These results support RL post-training as a broadly applicable approach to improving prediction quality and cross-seed stability in ACWMs. We will release all training and evaluation code and model checkpoints to support reproducibility and further research.

1 INTRODUCTION

Action-conditioned world models (ACWMs) predict how embodied environments evolve in response to robot actions. Given a history of visual observations and a prescribed action sequence, they generate future observations that capture the resulting robot motion, object interactions, and scene changes. By enabling robots to anticipate action consequences without executing every interaction in the real world, ACWMs provide a foundation for simulation, planning, and policy evaluation. Recent systems demonstrate the promise of this approach. Ctrl-World (Guo et al., 2026) supports learned simulation for policy post-training, IRASim (Zhu et al., 2025) enables model-based planning, and DreamDojo (Gao et al., 2026) and Genie Envisioner (Liao et al., 2026) support policy evaluation through imagined interactions.

A prominent approach to building ACWMs adapts large-scale pretrained video diffusion models, drawing on their rich priors over scene appearance, object motion, and temporal evolution. Stable Video Diffusion (Blattmann et al., 2023), Wan (Wan et al., 2025), Cosmos-Predict2.5 (Ali et al., 2025), and Cosmos3 (Agarwal et al., 2026) provide strong video generation capabilities spanning image-to-video (I2V), text-to-video (T2V), and physical-world modeling. However, whereas T2V and I2V conditions typically admit diverse plausible continuations, observed histories and prescribed actions in ACWM impose stronger constraints on the future. ACWM predictions are compared with observations collected after the prescribed actions are executed, assessing whether they accurately reproduce the resulting outcomes. *A plausible future alone is therefore insufficient; predictions must faithfully capture the effects of actions and the underlying real-world dynamics.*

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.

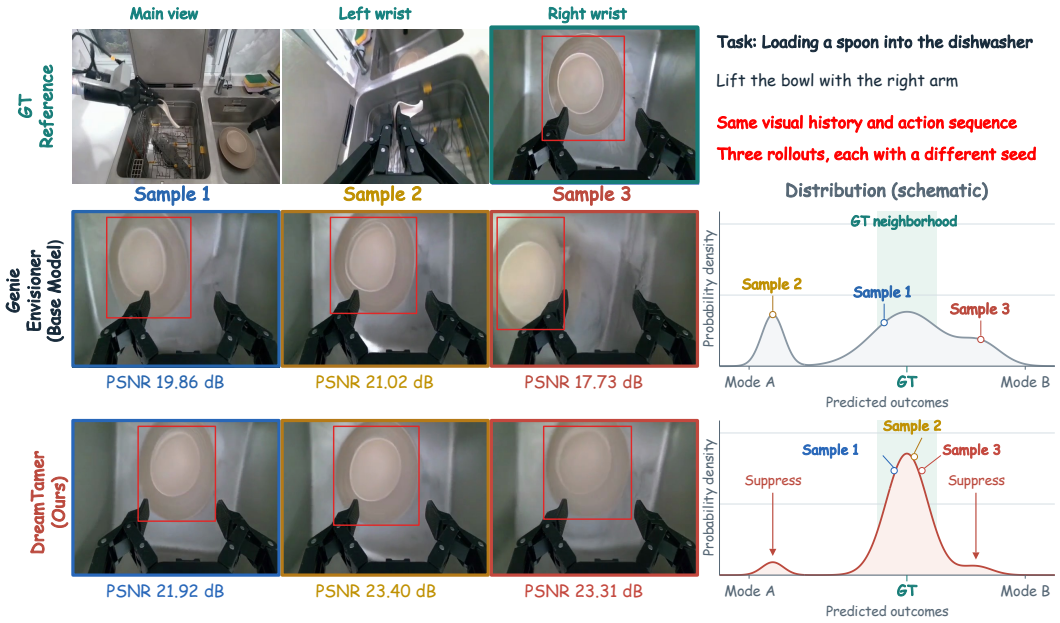


Fig. 1: **Reliable action-conditioned prediction across random seeds.** The robot lift a bowl with the right arm. Given the same visual history and action sequence, the base model produces inconsistent bowl locations in the right-wrist view when only the sampling seed changes (red boxes). Our method predicts the bowl's location more consistently across seeds and more accurately relative to the ground truth, illustrating improved fidelity beyond visual plausibility. The top row shows ground-truth observations from three camera views; the middle and bottom rows show three right wrist view from three rollouts of base model and our method at the same time instant. The schematic distributions illustrate the intended shift toward faithful predictions.

The requirement for consistently faithful action-conditioned prediction changes the role of repeated sampling. In creative generation, users can sample multiple candidates and select a preferred output through human judgment or learned reward models (Xu et al., 2023; 2026). For action-conditioned prediction, however, a visually convincing or preferred video is not necessarily a faithful simulation, and an inaccurate prediction may influence downstream decisions before its error is recognized. Reliability therefore cannot depend on resampling until a satisfactory video appears, especially when the true future is unavailable for verification. Yet our repeated-sampling experiments reveal substantial variation in prediction fidelity under identical observations and actions. As illustrated in Fig. 1, changing only the sampling seed can lead to different predicted interaction outcomes, whereas our method produces more consistent and faithful predictions. The sampling inconsistency raises the central question of our work. *How can we make action-conditioned predictions consistently faithful, rather than relying on repeated sampling to obtain an accurate future?*

The challenge is thus to favor faithful continuations and reduce low-fidelity predictions, rather than merely obtain an accurate future through repeated sampling. Reinforcement learning (RL) reshapes the prediction distribution through rewards. Methods such as DDPO (Black et al., 2024) and Flow-GRPO (Liu et al., 2026) enable this reward-guided post-training beyond the denoising objective. However, higher average reward alone does not establish that low-fidelity outcomes become less frequent. Our central goal is to establish whether RL post-training can systematically improve prediction consistency across distinct ACWM backbones while also improving fidelity. We introduce DreamTamer, a consistency-oriented RL post-training framework that targets unreliable predictions under fixed observation histories and action sequences. To target the stochastic decisions that most affect fidelity, we conduct denoising-stage branching experiments, which reveal greater sensitivity to early transitions, with the sensitive region varying across backbones. Drawing on temporal credit-assignment ideas (He et al., 2026; Li et al., 2026), we compare branches from shared denoising states and focus optimization on influential transitions. Across Ctrl-World, DreamDojo, and Genie Envisioner, DreamTamer improves both average and worst-of-four prediction quality while reduc-

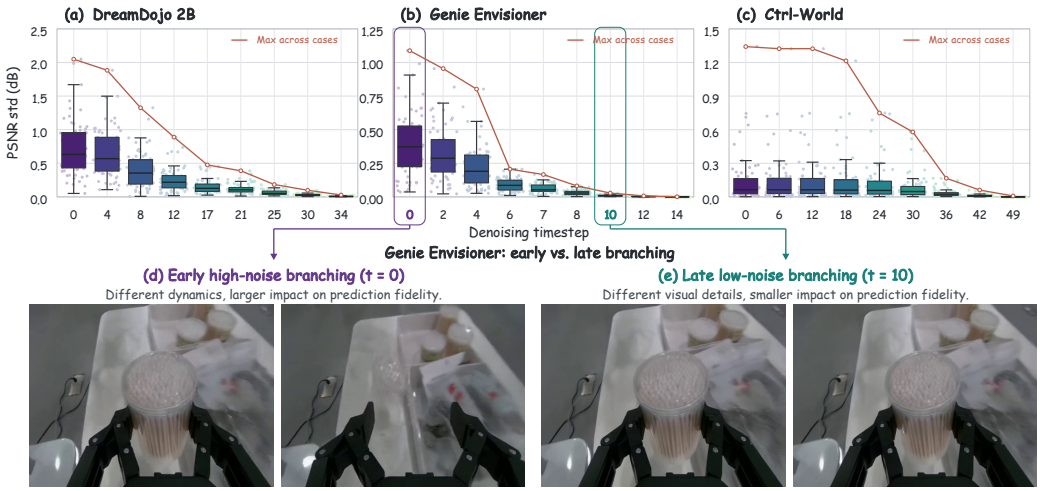


Fig. 2: **Early denoising stages capture task-critical interaction dynamics.** *Top:* At each sampled timestep, we branch the same intermediate state into three rollouts and compute their PSNR standard deviation (dots: cases; boxes: distributions; orange curves: maxima). Higher variance at early timesteps indicates a greater impact on prediction fidelity across all three models. *Bottom:* In Genie Envisioner (Liao et al., 2026), branching at timestep 0 changes the predicted interaction dynamics, whereas branching at timestep 10 preserves the interaction and mainly alters visual details.

ing variability across sampling seeds, demonstrating the broad effectiveness of RL post-training for consistently faithful action-conditioned prediction.

Our contributions are summarized as follows:

- **Consistency-oriented RL post-training across ACWMs.** We establish RL post-training as a broadly effective approach to improving the consistency of action-conditioned prediction. DreamTamer explicitly targets sampling-induced inconsistency under fixed observation histories and action sequences, improving prediction fidelity while reducing variability across seeds. Experiments on Ctrl-World, DreamDojo, and Genie Envisioner demonstrate these benefits across distinct ACWM backbones, including improvements in both average and worst-of-four prediction quality.
- **Uncovering dynamics-level sampling inconsistency in ACWMs.** We find that mainstream ACWMs produce inconsistent interaction outcomes across random seeds despite identical observations and actions. Denoising-stage branching experiments reveal a strong timestep bias. Early-stage stochasticity strongly affects prediction fidelity and task-critical interaction dynamics, whereas later-stage stochasticity mainly changes visual details.
- **Targeted optimization for consistently faithful prediction.** Guided by these findings, we develop dynamics-aware temporal credit assignment that localizes terminal fidelity differences to individual denoising transitions. We combine timestep-aware gradient correction and variance-aware sampling with a downside penalty that emphasizes low-fidelity predictions. Component ablations assess how these designs contribute to quality and consistency.

2 Related Work

Video Diffusion and Action-Conditioned World Models. Video diffusion models such as Stable Video Diffusion and Wan learn visual and motion priors from large-scale video data (Blattmann et al., 2023; Wan et al., 2025), while Cosmos extends video foundation models toward physical-world simulation and action modeling (Ali et al., 2025; Agarwal et al., 2026). Action-conditioned world models build on these priors to predict future observations from robot control inputs. Their applications include imitation-learning data augmentation with EVAC (Jiang et al., 2025), policy post-training with imagined rollouts in Ctrl-World (Guo et al., 2026), and reinforcement learning

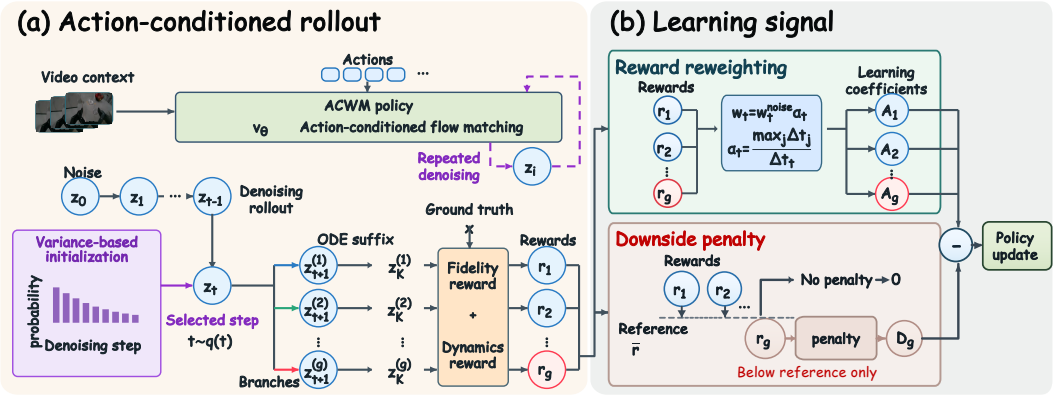


Fig. 3: Overview of DreamTamer. Denoising-stage branching provides transition-level credit for post-training, combining transition-scale correction, variance-initialized timestep sampling, and a one-sided downside penalty to improve prediction fidelity and sampling reliability.

within a video simulator in World4RL (Jiang et al., 2026). They also support model-based planning, as in IRASim (Zhu et al., 2025), and policy evaluation through imagined interactions, as in DreamDojo (Gao et al., 2026) and Genie Envisioner (Liao et al., 2026). Unlike open-ended video generation, these applications require faithful predictions of action consequences under fixed observation histories and action sequences.Our work therefore focuses on improving prediction fidelity and consistency across sampling seeds.

Reinforcement Learning for Diffusion and World Models. Reinforcement learning enables pretrained diffusion models to optimize task-specific rewards beyond their original denoising objectives. DDPO formulates denoising as a sequential decision process, while Flow-GRPO extends group-relative optimization to flow-based generation (Black et al., 2024; Liu et al., 2026). DiffusionNFT offers an alternative through reward-guided updates in the forward diffusion process (Zheng et al., 2026). These approaches reshape the sampling distribution to favor higher-reward outputs, providing a mechanism for improving action-conditioned prediction fidelity.

Related work explores noise-aware weighting and branching-based credit assignment for diffusion optimization (He et al., 2026; Li et al., 2026). Our controlled branching analysis identifies early denoising transitions as a major source of prediction variability in ACWMs. Building on this finding, our framework uses transition-level credit from the Flow-GRPO-fast branching implementation (Liu et al., 2026). We combine transition-scale correction with variance-initialized timestep sampling to adjust update magnitudes and allocate the branch-sampling budget across denoising stages. To improve reliability across sampling seeds, we further penalize rewards below the branch-group mean, amplifying negative credit while leaving above-mean centered rewards unchanged.

3 Method

3.1 Preliminaries

Action-conditioned denoising as a policy. Each training example is $(c, x^\star) \sim \mathcal{D}$, where $c = (h, \mathbf{u})$ contains a visual history and a prescribed robot action sequence, and $x^\star$ is the recorded future video. Given condition c, a model with parameters θ generates $\hat{x} = D(z_K)$ through K denoising transitions from an initial noisy latent z_0. We index denoising transitions by $k \in \{0, \dots, K-1\}$, separately from video-frame and robot control indices. The stochastic transition is modeled as (Black et al., 2024; Liu et al., 2026)

$$z_{k+1} \sim \pi_{\theta,k}(\cdot \mid z_k, c), \qquad \pi_{\theta,k} = \mathcal{N}\big(\mu_{\theta,k}(z_k, c),\, \nu_k^2 I\big)\,, \tag{1}$$

where ν_k is the policy sampling scale. The scheduler level is denoted by σ_k, with transition interval $\Delta t_k = \sigma_k - \sigma_{k+1}$; thus ν_k and Δt_k describe different quantities. For rollout i of condition n, $\xi_{n,i}$ denotes its sampling randomness and $\hat{x}_{n,i} = \mathrm{Rollout}_\theta(c_n; \xi_{n,i})$.

Group-relative policy optimization. A scalar reward $R(\hat{x}, x^\star; c)$ evaluates prediction fidelity, with larger values preferred. For each condition c_n, the behavior model θ_{old} samples a group of G rollouts with rewards $R_{n,1}, \ldots, R_{n,G}$. Let

$$\bar{R}_n = \frac{1}{G}\sum_{i=1}^{G} R_{n,i}, \qquad d_{n,i} = R_{n,i} - \bar{R}_n, \qquad v_n = \frac{1}{G-1}\sum_{i=1}^{G} d_{n,i}^2. \tag{2}$$

The standard group-relative coefficient is $A_{n,i}^{\text{grp}} = d_{n,i}/(\sqrt{v_n} + \varepsilon)$, where $\varepsilon > 0$ stabilizes normalization. For a sampled transition, define

$$\rho_{k,n,i}(\theta) = \frac{\pi_{\theta,k}(z_{n,k+1}^{(i)} \mid z_{n,k}^{(i)}, c_n)}{\pi_{\theta_{\text{old}},k}(z_{n,k+1}^{(i)} \mid z_{n,k}^{(i)}, c_n)}. \tag{3}$$

For controlled branching at stage k, all branches share the prefix state $z_{n,k}^{(i)} = z_{n,k}$. The clipped surrogate is (Liu et al., 2026)

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{n,i,k}\big[\, \min(\rho_{k,n,i}(\theta)A_{n,i}, \text{clip}(\rho_{k,n,i}(\theta), 1-\delta, 1+\delta)A_{n,i}) - \beta\,\mathcal{K}_{k,n,i}(\theta)\big], \tag{4}$$

where $A_{n,i}$ is either the standard coefficient above or a method-specific replacement, δ is the clipping range, and $\mathcal{K}_{k,n,i}$ is an optional KL regularizer to a frozen reference policy. The following sections modify this baseline through controlled branching, fixed variance-guided timestep sampling, dynamics-aware rewards, downside correction, and transition-scale weighting.

3.2 Localizing Sampling Sensitivity

To assess the pretrained model before RL post-training, we use controlled trajectory branching as in TempFlow-GRPO and Flow-GRPO-fast (He et al., 2026; Liu et al., 2026). For this diagnostic, θ denotes the frozen base-model parameters. For each condition c_n, we fix the initial noise and trajectory prefix to obtain a shared latent $z_{n,k}$. At stage k, we independently sample G transitions and propagate each resulting latent through a deterministic suffix:

$$z_{n,k+1}^{(i)} \sim \pi_{\theta,k}(\cdot \mid z_{n,k}, c_n), \quad \hat{x}_{n,k}^{(i)} = D\Big(\Phi_{\theta,k+1:K}(z_{n,k+1}^{(i)}; c_n)\Big), \quad i = 1, \ldots, G. \tag{5}$$

Here, $\Phi_{\theta,k+1:K}$ denotes the deterministic continuation from stage $k+1$ to the final latent; the condition and continuation rule are identical across branches.

Since the branches differ only in the sampled transition at stage k, their terminal reward differences estimate the effect of stochasticity at that stage, conditional on the fixed context and trajectory prefix.

Let $R_{n,k}^{(i)}$ be the terminal reward of branch i, and define the within-condition mean reward as $\bar{R}_{n,k} = G^{-1}\sum_i R_{n,k}^{(i)}$. We quantify stage-wise sensitivity by the mean standard deviation across conditions:

$$S(k) = \frac{1}{N}\sum_{n=1}^{N} s_{n,k}, \quad s_{n,k}^2 = \frac{1}{G-1}\sum_{i=1}^{G}\Big(R_{n,k}^{(i)} - \bar{R}_{n,k}\Big)^2. \tag{6}$$

For the diagnostic in Fig. 2, we instantiate the terminal fidelity measure R as PSNR

Fig. 2 shows that PSNR dispersion is concentrated in early denoising stages, falling from 0.726 to 0.008 dB for DreamDojo 2B (Gao et al., 2026) and from 0.385 to below 0.001 dB for Genie Envisioner (Liao et al., 2026), while Ctrl-World (Guo et al., 2026) maintains an approximately 0.145 dB plateau through timestep 18 before decreasing to 0.03 dB at timestep 36. The qualitative comparison in Fig. 2 is consistent with this trend, showing that branching at an early stage changes the predicted interaction, whereas branching later primarily affects visual details. For action-conditioned world modeling, this indicates that early denoising stages are more consequential for modeling action-induced dynamics, while later appearance refinements play a secondary role.

3.3 Method Design

DreamTamer (Fig. 3) uses the controlled trajectory-branching estimator described in Sec. 3.2, following the branching implementation of Flow-GRPO-fast (Liu et al., 2026). We combine this estimator with dynamics-aware rewards, transition-scale correction, variance-initialized timestep sampling, and a downside penalty to improve ACWM prediction fidelity and sampling reliability.

Reward design. We combine a video-similarity reward with a dynamics reward. The video-similarity reward $R_{\text{sim}}(\hat{x}_{n,i}, x_n^\star)$ follows the PersistWorld (Bardhan et al., 2026) evaluation protocol and measures agreement between the predicted and recorded videos. To assess action-induced motion, we adapt Foreground Displacement Chamfer Error (FDCE), introduced in (Wei et al., 2026). FDCE measures the discrepancy between generated and reference foreground displacement tracks using a symmetric Chamfer distance, with lower values indicating better motion agreement. The original evaluation protocol samples up to 16 valid foreground anchors per rollout pair; we reuse the points already selected for branching instead of sampling a separate subset. We denote this adapted FDCE by $E_{\text{disp}}(\hat{x}_{n,i}, x_n^\star; c_n)$, computed over the common visible region. We use the negative FDCE as our dynamics reward, $R_{\text{dyn}} = -E_{\text{disp}}$, and linearly combine it with the similarity reward as

$$R_{n,i} = R_{\text{sim}}(\hat{x}_{n,i}, x_n^\star) + \lambda_{\text{dyn}} R_{\text{dyn}}(\hat{x}_{n,i}, x_n^\star; c_n). \tag{7}$$

Transition-scale correction. Although early denoising transitions are most consequential for action-conditioned dynamics, their unscaled gradient norms under Flow-GRPO are comparatively small. Fig. 2 shows that PSNR dispersion is concentrated in early denoising stages, falling from 0.726 to 0.008 dB for DreamDojo 2B (Gao et al., 2026) and from 0.385 to below 0.001 dB for Genie Envisioner (Liao et al., 2026), while Ctrl-World (Guo et al., 2026) maintains an approximately 0.145 dB plateau through timestep 18 before decreasing to 0.03 dB at timestep 36. The qualitative comparison in Fig. 2 is consistent with this trend, showing that branching at an early stage changes the predicted interaction, whereas branching later primarily affects visual details. Our gradient scaling compensates for this imbalance by increasing the relative contribution of early denoising transitions to the policy update.

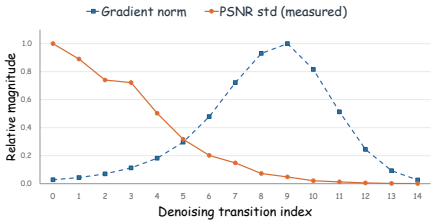


Fig. 4: Gradient norm and PSNR dispersion across denoising stages.

Variance-initialized timestep sampling. Transition-scale correction adjusts update magnitudes, but it does not imply that every denoising stage deserves the same branch-sampling budget. We observe that stages with larger reward dispersion produce more variable credit signals, suggesting that additional samples at these stages are more informative. Under a fixed branch budget, we therefore initialize the timestep distribution from the base model's reward dispersion:

$$q_k = \frac{S(k)}{\sum_{j=0}^{K-1} S(j)}. \tag{8}$$

This variance-guided allocation concentrates computation on stages with more informative credit signals. Prop. A.2 and its connection to $S(k)$ in App. A.4 give the allocation intuition; we compute the distribution once before training and keep it fixed thereafter. In our ablation, this initialization improves reward more rapidly than uniform timestep sampling.

Downside penalty. For an ACWM, repeated predictions under the same visual history and action condition should remain faithful to the prescribed action consequences. This makes low-fidelity outcomes especially undesirable: a large downside deviation indicates an unreliable action-conditioned prediction. We initially considered penalizing reward deviations symmetrically, but found that penalizing favorable deviations as well suppresses useful high-reward rollouts and degrades training. We therefore use a one-sided downside penalty. For each group, let $d_{n,i} = R_{n,i} - \bar{R}_n$ be the centered reward, and define

$$A_{n,i} = \gamma_{\text{adv}} \left[d_{n,i} - \lambda_{\text{eff}} [\min(d_{n,i}, 0)]^2 \right], \quad \gamma_{\text{adv}} = \text{scale} \, \frac{G}{G-1}. \tag{9}$$

Here $\gamma_{\text{adv}} > 0$ is the fixed advantage scale; c_n continues to denote the conditioning context. When a rollout is above the group mean, its original positive reward is unchanged while it falls below the mean, its negative credit is amplified according to the shortfall. This preserves favorable training signals while more strongly suppressing low-fidelity outcomes. Our ablation study shows that this design reduces reward variance across repeated samples of the same condition.

Table 1: Prediction quality and sampling reliability on Genie Envisioner.

	Main				Wrist			Overall		
A. Prediction quality (mean)										
Model	PSNR ↑	LPIPS ↓	SSIM ↑	FDCE ↓	PSNR ↑	LPIPS ↓	SSIM ↑	PSNR ↑	LPIPS ↓	SSIM ↑
Genie-Envisioner	25.44	0.063	0.881	2.799	23.63	0.141	0.805	24.23	0.115	0.830
DreamTamer (Ours)	**26.84**	**0.056**	**0.894**	**1.974**	**25.06**	**0.121**	**0.831**	**25.65**	**0.100**	**0.852**
B. Worst-of-four prediction quality										
Model	PSNR ↑	LPIPS ↓	SSIM ↑	FDCE ↓	PSNR ↑	LPIPS ↓	SSIM ↑	PSNR ↑	LPIPS ↓	SSIM ↑
Genie-Envisioner	25.04	0.067	0.877	3.226	23.06	0.154	0.793	23.72	0.125	0.821
DreamTamer (Ours)	**26.58**	**0.058**	**0.892**	**2.172**	**24.77**	**0.125**	**0.825**	**25.37**	**0.103**	**0.847**
C. Sampling variance reduction (%; ↑ better)										
Comparison	PSNR	LPIPS	SSIM	FDCE	PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM
vs. Genie-Envisioner	64.43	84.50	72.43	82.74	70.97	86.19	71.81	69.70	86.67	73.77

4 EXPERIMENTS

4.1 MAIN RESULTS

Datasets and base models. We evaluate our post-training framework on two datasets and three pretrained action-conditioned world models, covering both single-view and three-view prediction. On the AgiWorld Alpha dataset (Bu et al., 2025), we randomly sample 1,001 examples across tasks to form the validation set, and use remaining examples for training. We use DreamDojo (Gao et al., 2026) for single-view prediction and Genie Envisioner (Liao et al., 2026) for three-view prediction. On DROID (Khazatsky et al., 2024), we use Ctrl-World (Guo et al., 2026) with three camera views and follow the data and training settings of PersistWorld (Bardhan et al., 2026).

Evaluation protocol. We evaluate each condition with **four random seeds**, holding the visual history and action sequence fixed, to measure both prediction quality and sampling variability. We compare generated videos with ground-truth continuations using PSNR and SSIM (Hore & Ziou, 2010), and LPIPS (Zhang et al., 2018). On AgiWorld Alpha, we also report Foreground Displacement Chamfer Error (FDCE) (Wei et al., 2026) for foreground motion agreement (App. A.2). We average per-condition seed means and variances over the evaluation set, with view-specific breakdowns where applicable. Worst-of-four quality for Genie Envisioner and Ctrl-World, and visual worst-of-four quality for DreamDojo, is evaluated by selecting each model's lowest-joint-reward seed within each condition. The corresponding metrics are read from that same prediction and then averaged over matched conditions. These reward-selected scores need not be the least favorable values of each individual metric. DreamDojo FDCE instead reports the maximum error over four seeds, and the component ablations report the maximum LPIPS. DreamDojo FDCE means, variances, and worst-of-four scores use the same 998 single-rollout and 970 long-rollout conditions with complete four-seed scores in both models, without an additional track-count threshold. Subset selection and evaluation details are provided in App. A.2. In Tables 1–3, Panels A and B report mean and worst-of-four quality, and Panel C reports variance reduction relative to the indicated baseline. Negative reductions indicate increased variance; bold marks the best scores in Panels A and B, including ties at the displayed precision. Qualitative comparisons for all three models appear in App. A.1

Results on Genie Envisioner. DreamTamer improves mean and worst-of-four prediction quality across head and wrist views (Table 1). The most pronounced stability gain is an 86.67% reduction in overall LPIPS variance; mean head-view FDCE also decreases by 29.48%, indicating better motion agreement. These gains accompany improvements in the remaining visual metrics and lower sampling variability. The reward-selected worst-of-four scores even outperform the base model's mean scores across all reported metrics. Together, the improvements in visual reconstruction and foreground motion suggest that post-training benefits both appearance fidelity and action-conditioned dynamics across repeated predictions. Comparison visualizations are provided in the appendix.

Table 2: Prediction quality and sampling reliability on DreamDojo

	Single rollout				Long rollout			
Model	PSNR ↑	LPIPS ↓	SSIM ↑	FDCE ↓	PSNR ↑	LPIPS ↓	SSIM ↑	FDCE ↓
A. Prediction quality (mean)								
DreamDojo	21.68	0.121	0.836	9.670	18.92	0.196	0.772	17.780
DreamTamer (Ours)	**22.62**	**0.106**	**0.851**	**7.985**	**19.91**	**0.172**	**0.799**	**14.819**
B. Worst-of-four prediction quality								
DreamDojo	21.04	0.131	0.826	11.559	18.47	0.208	0.760	22.105
DreamTamer (Ours)	**22.05**	**0.112**	**0.846**	**9.770**	**19.57**	**0.178**	**0.795**	**18.627**
C. Sampling variance reduction (%; ↑ better)								
Comparison	PSNR	LPIPS	SSIM	FDCE	PSNR	LPIPS	SSIM	FDCE
vs. DreamDojo	32.12	69.36	80.17	6.19	48.25	73.95	84.96	34.67

Table 3: Prediction quality and sampling reliability on Ctrl-World, compared with PersistWorld.

	External			Wrist			Overall		
A. Prediction quality (mean)									
Model	PSNR ↑	LPIPS ↓	SSIM ↑	PSNR ↑	LPIPS ↓	SSIM ↑	PSNR ↑	LPIPS ↓	SSIM ↑
Ctrl-World	23.40	0.082	0.833	18.60	0.295	0.634	21.80	0.153	0.767
PersistWorld	24.16	0.074	0.849	**19.34**	0.286	**0.666**	22.55	0.145	0.788
DreamTamer (Ours)	**24.40**	**0.072**	**0.858**	19.18	**0.277**	**0.666**	**22.66**	**0.139**	**0.794**
B. Worst-of-four prediction quality									
Model	PSNR ↑	LPIPS ↓	SSIM ↑	PSNR ↑	LPIPS ↓	SSIM ↑	PSNR ↑	LPIPS ↓	SSIM ↑
Ctrl-World	23.10	0.086	0.827	18.21	0.307	0.623	21.47	0.160	0.759
PersistWorld	23.97	0.076	0.846	19.06	0.293	0.659	22.33	0.148	0.784
DreamTamer (Ours)	**24.13**	**0.073**	**0.852**	**19.09**	**0.282**	**0.661**	**22.42**	**0.143**	**0.790**
C. Sampling variance reduction (%; ↑ better)									
Comparison	PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM
vs. Ctrl-World	56.36	89.03	86.39	55.66	55.42	64.18	59.38	75.78	80.93
vs. PersistWorld	10.88	25.50	40.35	13.49	-1.52	20.02	9.87	11.81	30.25

Results on DreamDojo. DreamTamer improves mean and worst-of-four visual quality and FDCE at both rollout horizons (Table 2). Mean FDCE decreases by 17.42% for single rollouts and 16.65% for long rollouts, while FDCE seed variance decreases by 6.19% and 34.67%, respectively. The largest variance reduction is 84.96% for long-rollout SSIM, with lower variance across all remaining metrics. Visual quality gains persist in long rollouts, where absolute SSIM and LPIPS improvements are larger than in single rollouts. FDCE statistics use matched conditions with all four seeds available for both models, without an additional track-count threshold. Worst-of-four FDCE is the maximum error over the four seeds (App. A.2).

Results on Ctrl-World. DreamTamer improves all three overall mean and worst-of-four metrics over Ctrl-World and PersistWorld (Table 3). Compared with PersistWorld, it produces more stable predictions across seeds, with overall SSIM variance decreasing by 30.25% and lower overall PSNR and LPIPS variance. The corresponding SSIM variance reduction over Ctrl-World is 80.93%. These stability gains accompany better external-view quality, although wrist-view mean PSNR remains below PersistWorld and wrist LPIPS variance increases slightly. The comparison with PersistWorld shows that DreamTamer improves sampling reliability even against a post-trained baseline, while retaining gains in overall prediction quality. Comparison visualizations are provided in the appendix.

Table 4: Optimization and reward ablations.

(a) Components: Genie Envisioner.

Variant	Mean ↓	Worst ↓	Var. (1e-5) ↓
Base model	0.115	0.125	10.8
Full method	0.106	**0.113**	**5.4**
w/o scale correction	0.112	0.120	8.8
w/o Var. sampling	0.109	0.118	6.6
w/o downside penalty	**0.105**	0.115	5.9

(b) Reward design.

Model / reward	PSNR ↑	SSIM ↑	LPIPS ↓	FDCE ↓
Genie Envisioner	24.23	0.830	0.115	2.80
RL (Sim.)	**24.95**	0.838	**0.106**	2.52
RL (Sim. + Dyn.)	24.92	**0.840**	**0.106**	**2.35**
DreamDojo	21.68	0.836	0.121	9.67
RL (Sim.)	**22.29**	**0.846**	**0.116**	**8.91**
RL (Sim. + Dyn.)	19.91	0.797	0.140	10.64

4.2 Ablation Study

We evaluate the optimization components on Genie Envisioner (Liao et al., 2026) and examine reward design on both Genie Envisioner and DreamDojo (Gao et al., 2026). Table 4 evaluates optimization components and reward choices against pretrained baselines. The ablations on both Genie Envisioner and DreamDojo use checkpoints saved after 200 training steps. We remove one component at a time with the reward fixed; Var. sampling denotes variance-based timestep sampling. Mean, Worst, and Var. summarize balanced LPIPS as the four-seed mean, maximum, and variance, averaged over conditions. Reward ablations retain all components and compare RL with similarity reward alone (Sim.) and with the added dynamics reward (Sim. + Dyn.). We report mean visual metrics and head-view FDCE for Genie Envisioner, and single-rollout mean metrics for DreamDojo; model-name rows denote pretrained baselines. The Genie Envisioner ablations retain the original balanced visual metrics; the main results instead report equal-weight averages across views.

Optimization components. Table 4a shows that the full method reduces mean LPIPS (Hore & Ziou, 2010) from 0.115 to 0.106 and worst-of-four LPIPS from 0.125 to 0.113, while reducing seed variance from 10.8×10^{-5} to 5.4×10^{-5}, a 50.0% reduction at the reported precision. Removing scale correction gives the highest mean LPIPS, worst-of-four LPIPS, and variance among the three component ablations. Removing variance-based sampling increases worst-of-four LPIPS from 0.113 to 0.118 and seed variance from 5.4×10^{-5} to 6.6×10^{-5}. The downside penalty reduces worst-of-four LPIPS from 0.115 to 0.113 and seed variance from 5.9×10^{-5} to 5.4×10^{-5}, although mean LPIPS increases slightly from 0.105 to 0.106. It therefore improves worst-seed performance and reduces variation across seeds at a small cost to mean perceptual quality.

Reward design. On Genie Envisioner (Liao et al., 2026), both reward settings improve all four metrics over the base model (Table 4b). Adding the dynamics reward reduces FDCE from 2.52 to 2.35 (approximately 6.7%) and raises SSIM from 0.838 to 0.840. LPIPS remains at 0.106 at the reported precision, while PSNR decreases from 24.95 to 24.92 dB. Thus, the improvement in motion agreement comes with a small decrease in pixel-level fidelity.

On DreamDojo (Gao et al., 2026), adding the FDCE-based dynamics reward causes training to collapse, with all four short-horizon metrics falling below the pretrained baseline (Table 4b). Poor action following in base model leads to large fluctuations in FDCE, producing an unstable reward signal that destabilizes RL training. We use the dynamics reward only for base models with reliable action following, so DreamDojo uses only the similarity reward in our main experiments (App. A.2).

5 Conclusion

In this work, we are, to our knowledge, the first to explicitly formulate cross-seed prediction inconsistency as a problem in ACWMs and develop a general RL post-training framework to address it. Guided by an analysis of ACWM denoising dynamics, DreamTamer introduces transition-level credit assignment, scale correction, variance guided timestep sampling and downside penalty to produce more consistently faithful predictions. Experiments on three distinct ACWM backbones demonstrate improved prediction quality and cross-seed consistency, supporting the effectiveness and broad applicability of our approach.

AI USE STATEMENT

We used AI tools to assist with language polishing and revisions to improve the clarity and readability of the manuscript. The authors take full responsibility for the scientific content and final wording of the paper.

REFERENCES

Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, et al. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026.

Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, et al. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 2025.

Jai Bardhan, Patrik Drozdik, Josef Sivic, and Vladimir Petrik. Persistent robot world models: Stabilizing multi-step rollouts via reinforcement learning. *arXiv preprint arXiv:2603.25685*, 2026.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, volume 2024, pp. 4965–4987, 2024.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, et al. Dreamdojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026.

Yanjiang Guo, Lucy Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 6121–6138, 2026.

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. In *International Conference on Learning Representations*, volume 2026, pp. 132869–132895, 2026.

Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pp. 2366–2369. IEEE, 2010.

Yuxin Jiang, Shengcong Chen, Siyuan Huang, Liliang Chen, Pengfei Zhou, Yue Liao, Xindong He, Chiming Liu, Hongsheng Li, Maoqing Yao, et al. Enerverse-ac: Envisioning embodied environments with action condition. *arXiv preprint arXiv:2505.09723*, 2025.

Zhennan Jiang, Kai Liu, Yuxing Qin, Shuai Tian, Yupeng Zheng, Mingcai Zhou, Zhengtao Zhang, Chao Yu, Haoran Li, and Dongbin Zhao. World4rl: Diffusion world models for policy refinement with reinforcement learning for robotic manipulation. *IEEE Robotics and Automation Letters*, 2026.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

Yuming Li, Yikai Wang, Zhongyu Zhao, Ming Lu, Qi She, Shanghang Zhang, et al. Branchgrpo: Stable and efficient grpo with structured branching in diffusion models. In *International Conference on Learning Representations*, volume 2026, pp. 20839–20866, 2026.

Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Si Liu, Jianlan Luo, Liliang Chen, et al. Genie envisioner: A unified world foundation platform for robotic manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 88446–88463, 2026.

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. *Advances in neural information processing systems*, 38:40783–40818, 2026.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Yufan Wei, Kun Zhou, Lingjun Mao, Zijun Zhang, Ziming Xu, Ziqiao Xi, Shuang Liang, Ruobing Han, Yuchen Yan, Xinyue Wang, et al. Causally debiased latent action model for embodied action conditioned world models. *arXiv preprint arXiv:2607.09185*, 2026.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

Jiazheng Xu, Yu Huang, Jiale Cheng, Yuanming Yang, Jiajun Xu, Yuan Wang, Wenbo Duan, Shen Yang, Qunlin Jin, Shurun Li, et al. Visionreward: Fine-grained multi-dimensional human preference learning for image and video generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11269–11277, 2026.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

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.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: A fine-grained world model for robot manipulation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9834–9844. IEEE, 2025.

A Appendix

A.1 Additional Comparison Visualizations

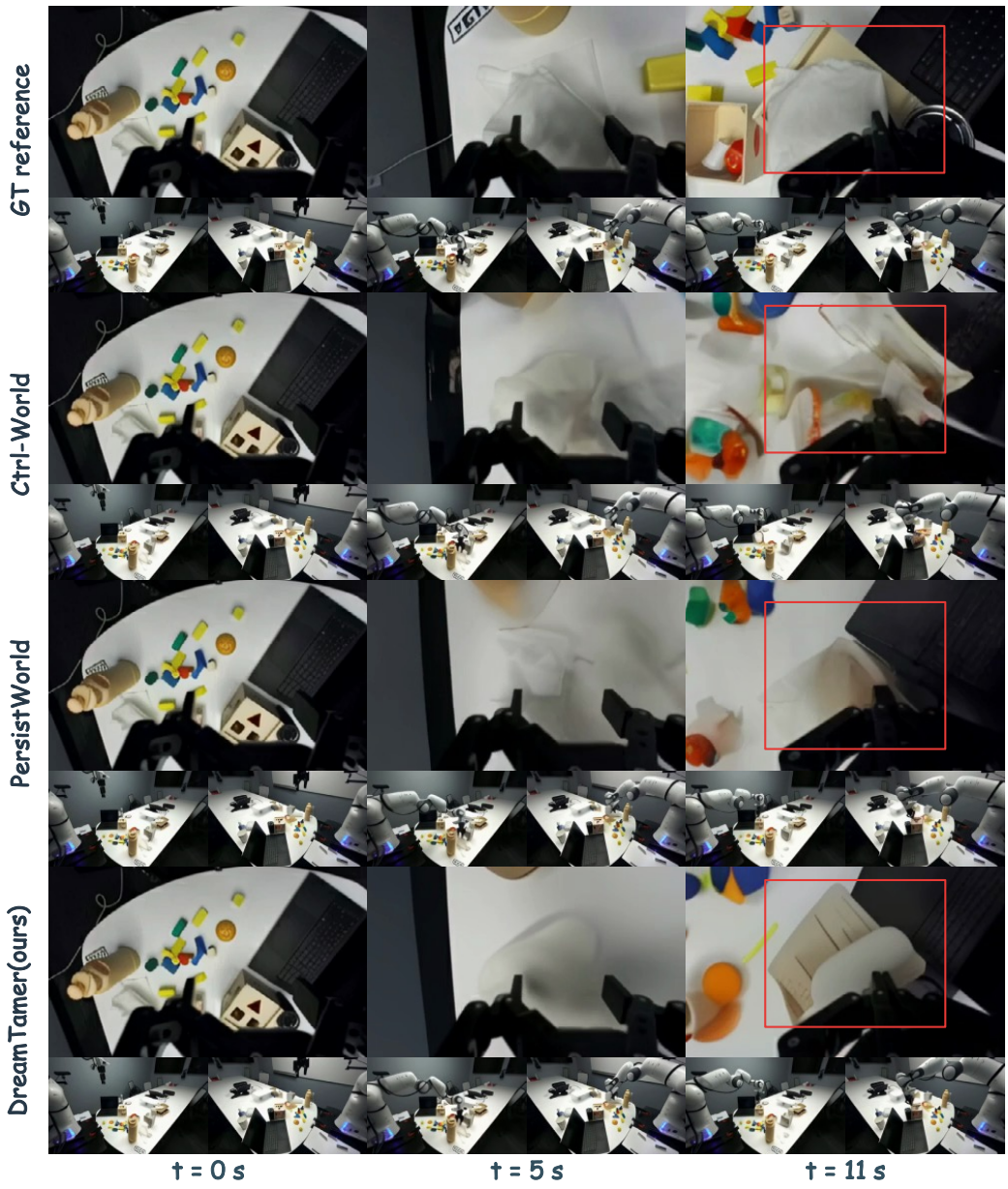


Fig. 5: **Multi-view prediction comparison on Ctrl-World.** Row groups show the ground-truth reference, the pretrained Ctrl-World (Guo et al., 2026) model, PersistWorld (Bardhan et al., 2026), and DreamTamer, respectively. Columns show the scene at $t = 0$, 5, and 11 s. Each group contains a large wrist-view image and two smaller external-camera views, allowing the local interaction to be compared with the robot's motion in the surrounding scene. The reference sequence shows the robot manipulating a tissue near a box. Both Ctrl-World and PersistWorld fail to reproduce the successful tissue manipulation and the box geometry, with blurred and structurally distorted predictions in the interaction region. DreamTamer more faithfully reproduces the successful interaction while preserving a clearer tissue and box configuration. Red boxes highlight these differences at the final time point, illustrating improvements in both action-conditioned dynamics and visual fidelity.

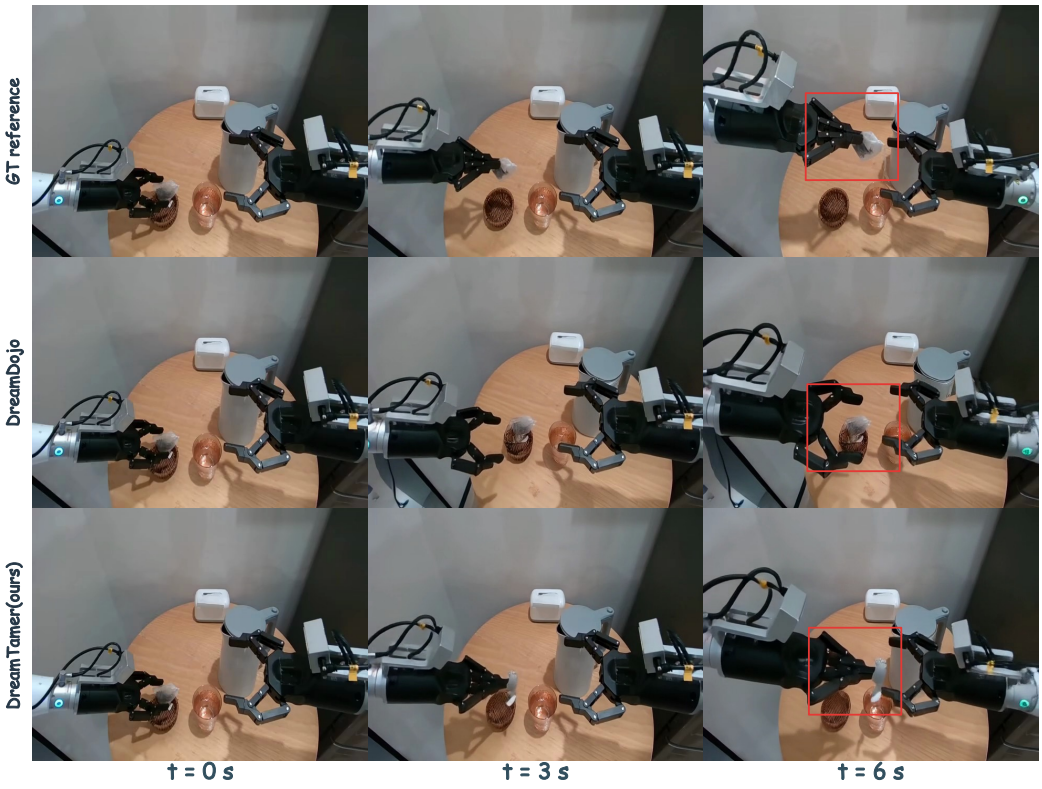


Fig. 6: **Prediction of tea-bag pickup on DreamDojo.** Rows show the ground-truth reference, the pretrained DreamDojo (Gao et al., 2026) model, and DreamTamer, respectively, with columns corresponding to $t = 0$, 3, and 6 s. In the reference sequence, the left gripper picks up a tea bag from a small container while the right gripper remains near the pitcher. The base model follows the prescribed actions inaccurately and fails to reproduce the pickup, leaving the tea bag near the container as the left gripper moves away. DreamTamer follows the actions more accurately and correctly predicts the tea bag being lifted by the left gripper. Red boxes highlight the final gripper–tea-bag configuration, showing how improved action following leads to a more faithful prediction of the manipulation outcome.



Fig. 7: **Multi-view prediction of grasping and lifting an orange on Genie Envisioner.** Row groups show the ground-truth reference, the pretrained Genie Envisioner (Liao et al., 2026) model, and DreamTamer, respectively. Columns correspond to $t = 0$, 2, and 4 s, and each group includes a large wrist-view image with two smaller views providing scene context. The reference sequence shows the gripper closing around an orange and lifting it. The base model inaccurately predicts the gripper motion, failing to close the fingers around the orange, which consequently remains unlifted in the predicted sequence. DreamTamer more accurately reproduces the grasp and subsequent lift, with the orange held between the fingers as the gripper moves. Red boxes highlight the final gripper–orange configuration, making the difference between the failed and successful predicted grasps visible.

A.2 Foreground Displacement Chamfer Error

Definition A.1 (Foreground Displacement Chamfer Error)**.** FDCE (Wei et al., 2026) compares foreground motion in generated and reference videos. Let $\hat{p}_i^t$ and p_j^t be valid foreground point tracks in the generated and reference videos, respectively, over a common window with anchor frame $t = 0$ and H subsequent frames. Define

$$\hat{a}_i^t = \hat{p}_i^t - \hat{p}_i^0, \qquad a_j^t = p_j^t - p_j^0, \qquad c_{ij} = \frac{1}{H} \sum_{t=1}^{H} \|\hat{a}_i^t - a_j^t\|_2. \tag{10}$$

For N_g generated tracks and N_r reference tracks, the metric is

$$\mathrm{FDCE} = \frac{1}{2N_g} \sum_{i=1}^{N_g} \min_{1 \le j \le N_r} c_{ij} + \frac{1}{2N_r} \sum_{j=1}^{N_r} \min_{1 \le i \le N_g} c_{ij}. \tag{11}$$

The distance is measured in pixels at the evaluation resolution. It compares motion relative to each track's anchor rather than absolute point locations, with bidirectional nearest-neighbor matching allowing different numbers of valid tracks.

The original protocol selects up to 16 valid foreground anchors per video pair. Our training reward adapts this displacement distance by reusing rollout-selected points, as described in the reward design paragraph. The FDCE evaluation score is the displacement error itself, rather than its reward transformation.

Dynamics reward settings. In the main experiments, we use the dynamics reward R_{dyn} in Eq. (7) only for Genie Envisioner (Liao et al., 2026). DreamDojo (Gao et al., 2026) and Ctrl-World (Guo et al., 2026) use only the video-similarity reward R_{sim}, with $\lambda_{\mathrm{dyn}} = 0$. When a base model does not reliably follow the prescribed actions, the displacement-based reward can be noisy and destabilize post-training. We therefore restrict its use to settings with sufficient action-following fidelity. This choice concerns the training reward; FDCE is still reported as an evaluation metric where specified.

Shared evaluation protocol. For all three models, we evaluate each condition with four fixed seeds, holding the observation history and action sequence fixed. We compute per-condition seed means and sample variances (denominator $4 - 1 = 3$), then average over matched conditions. For worst-of-four results, each model selects its lowest-joint-reward seed within each condition and reports all metrics and views from that prediction, followed by averaging over conditions, except for DreamDojo FDCE, which uses the maximum error across the four seeds. Variance reduction is $100(1-V_{\mathrm{ours}}/V_{\mathrm{baseline}})$, where V is the mean within-condition variance and the baseline is indicated in the table. For Genie Envisioner and Ctrl-World, Overall visual metrics average the three views equally for each prediction before computing seed statistics, retaining cross-view covariance in the overall variance. Grouped views (Wrist or External) instead average the corresponding view-specific means, variances, and worst-of-four scores. Their variance reductions use the ratio of these averaged variances.

Genie Envisioner evaluation details. We fine-tune Genie Envisioner (Liao et al., 2026) for 700 training steps using LoRA with rank 64 and a learning rate of 4×10^{-5}.

We evaluate both the pretrained base model and DreamTamer on all 1,001 conditions with four seeds per condition. For one condition, the ground-truth reference itself contains no eligible tracks under the evaluation rule, so FDCE is undefined for every seed in both models. This exclusion reflects missing trajectory support in the reference rather than a model prediction failure. Reported metrics are aggregated over the remaining 1,000 matched complete conditions (4,000 predictions per model). Wrist combines the left and right wrist views, and FDCE is evaluated on the head view only.

Ctrl-World evaluation details. We fine-tune Ctrl-World (Guo et al., 2026) following the training protocol of PersistWorld (Bardhan et al., 2026), with a maximum of 8,000 training steps, and report the model with the best validation performance during training. The pretrained base model, official PersistWorld model, and our trained model are evaluated on 256 matched conditions using complete closed-loop rollouts. External combines the two external views.

Table 5: Causes of missing DreamDojo FDCE scores after recomputation. Single and long rollouts correspond to Short and Rollout, respectively. Counts are pooled across both models and all seeds. A missing score indicates that no tracks satisfy the evaluation rule, not necessarily that the predicted action is incorrect.

Cause	Short	Rollout	Total
No eligible tracks in the GT reference	8	24	32
Eligible GT tracks, but none in the prediction	14	41	55
Eligible tracks on both sides, but no valid intersection at corresponding points and frames	0	39	39
Total	22	104	126

DreamDojo evaluation details. We fine-tune DreamDojo (Gao et al., 2026) for 600 training steps, using the same learning rate and LoRA rank as Genie Envisioner.

Single and long rollouts contain 13 frames (one clip) and 49 frames (four clips), respectively. Visual means and variances use all 1,001 conditions at each horizon. Visual worst-of-four scores use conditions with four numeric joint rewards in both models, leaving 998 single-rollout and 970 long-rollout conditions.

DreamDojo FDCE evaluation after completion of missing scores. Following completion of missing FDCE scores, we compare the models on 998 single-rollout and 970 long-rollout conditions with all four seed scores available for both models. We impose no additional minimum of four or eight common tracks. For each condition, we compute the four-seed mean, maximum FDCE, and sample variance with denominator $4 - 1 = 3$, then average each statistic over conditions. Thus, worst-of-four FDCE is a metric-specific maximum, whereas visual worst-of-four metrics retain the lowest-joint-reward seed definition. Predictions without a computable FDCE remain unscored.

Mean FDCE decreases from 9.670012 to 7.985081 for single rollouts (17.42%) and from 17.779683 to 14.819278 for long rollouts (16.65%). Worst-of-four FDCE decreases from 11.558964 to 9.769580 (15.48%) and from 22.104956 to 18.626870 (15.73%), respectively. Mean within-condition seed variance decreases from 11.264482 to 10.566988 (6.19%) for single rollouts and from 65.562474 to 42.830373 (34.67%) for long rollouts.

Why some predictions have no FDCE score. Under the common-visibility rule, filtering can retain zero eligible tracks, leaving FDCE undefined even after recomputation. This does not by itself indicate incorrect actions or a complete failure to detect the robot arm. Table 5 separates three causes according to whether eligible tracks exist in the ground-truth reference, the prediction, and their intersection at corresponding points and frames. The counts refer to missing model–condition–seed scores pooled across Base and DreamTamer, rather than distinct conditions.

Why the main results use matched samples. A condition is included only if all eight FDCE scores are available, comprising four seeds for Base and the same four seeds for DreamTamer. If any seed is missing for either model, the entire condition is excluded from both models' statistics. This ensures that the models are compared on identical samples and that worst-of-four scores and seed variances are computed with the same number of seeds. Means, worst-of-four scores, and variances consequently share the same comparison set. Matching conditions across models is distinct from the common visible tracks and frames used within each prediction–reference pair. These results apply to the matched complete conditions, with no scores imputed for predictions whose FDCE is undefined.

A.3 Derivation of Transition-Scale Correction

We derive the gradient of the branch objective $\rho_{k,n,i}(\theta)A_{n,i}$ at $\theta = \theta_{\text{old}}$ for the Gaussian transition policy in Eq. (1). The derivation identifies the factor Δt_k contributed by the transition mean and yields the weighted gradient in Eq. (16).

Fix a condition c_n, a stage $k \in \{0, \dots, K-1\}$, its prefix state $z_{n,k}$, and a sampled branch i. Let $\Delta t_k = \sigma_k - \sigma_{k+1} > 0$. The Gaussian policy in Eq. (1) has standard deviation $\nu_k > 0$ and mean

$$\mu_{\theta,k}(z_{n,k}, c_n) = z_{n,k} + \Delta t_k b_{\theta,k}(z_{n,k}, c_n), \tag{12}$$

where $b_{\theta,k}$ is the sampler drift, including any deterministic SDE correction, and is differentiable in θ. The behavior policy with parameters θ_{old} generates the branch as

$$z^{(i)}_{n,k+1} = \mu_{\theta_{\text{old}},k}(z_{n,k}, c_n) + \nu_k \epsilon^{(i)}_{n,k}, \qquad \epsilon^{(i)}_{n,k} \sim \mathcal{N}(0, I_{d_z}), \tag{13}$$

with d_z the latent dimension. During differentiation, the condition, prefix, sampled branch, scheduler, and ν_k are fixed. The branch coefficient $A_{n,i}$ from Eq. (9) is also held fixed; its stage index is omitted as in the main text.

Prop. A.1 (Local transition scaling)**.** *Let $\ell_{k,n,i}(\theta) = \rho_{k,n,i}(\theta) A_{n,i}$, where $\rho_{k,n,i}$ is the likelihood ratio in Eq.* (3)*, and define the drift Jacobian*

$$B_{n,k} = \left.\frac{\partial b_{\theta,k}(z_{n,k}, c_n)}{\partial \theta}\right|_{\theta=\theta_{\text{old}}}.$$

Under the conditions above, the branch gradient at the behavior parameters is

$$g_{k,n,i} := \nabla_\theta \ell_{k,n,i}(\theta)|_{\theta=\theta_{\text{old}}} = \frac{\Delta t_k}{\nu_k} A_{n,i} B^{\mathsf{T}}_{n,k} \epsilon^{(i)}_{n,k}. \tag{14}$$

Following the noise-aware policy weighting of TempFlow-GRPO (He et al., 2026), we weight each transition by its injected noise standard deviation ν_k, normalized to have unit mean over the K denoising transitions. Define

$$\bar{\nu} = \frac{1}{K}\sum_{j=0}^{K-1} \nu_j, \qquad w_k^{\text{noise}} = \frac{\nu_k}{\bar{\nu}}, \qquad \frac{1}{K}\sum_{k=0}^{K-1} w_k^{\text{noise}} = 1. \tag{15}$$

Here ν_k is the full standard deviation of the Gaussian transition, including the step-size contribution to the injected noise. The noise schedule is fixed with respect to θ, so these weights are also fixed during differentiation. In addition to this noise weighting, we apply the step-size correction $\Delta t_{\max}/\Delta t_k$, where $\Delta t_{\max} = \max_{0 \le j < K} \Delta t_j$. The combined weight $w_k = w_k^{\text{noise}} \Delta t_{\max}/\Delta t_k$ gives

$$w_k g_{k,n,i} = \Delta t_{\max} \frac{w_k^{\text{noise}}}{\nu_k} A_{n,i} B^{\mathsf{T}}_{n,k} \epsilon^{(i)}_{n,k}. \tag{16}$$

Proof. We expand the Gaussian density as in TempFlow-GRPO (He et al., 2026). For readability, write $y = z^{(i)}_{n,k+1}$, $\mu_\theta = \mu_{\theta,k}(z_{n,k}, c_n)$, $\pi_\theta = \pi_{\theta,k}(y \mid z_{n,k}, c_n)$, and $J_\theta = \partial \mu_\theta / \partial \theta$. All gradients are column vectors. The covariance matrix is $\Sigma_k = \nu_k^2 I_{d_z}$, so

$$|\Sigma_k| = (\nu_k^2)^{d_z} = \nu_k^{2d_z}, \qquad \Sigma_k^{-1} = \nu_k^{-2} I_{d_z}.$$

Substitution into the Gaussian density gives

$$\begin{aligned}
\pi_\theta &= (2\pi)^{-d_z/2} |\Sigma_k|^{-1/2} \exp\left[-\tfrac{1}{2}(y-\mu_\theta)^{\mathsf{T}} \Sigma_k^{-1} (y-\mu_\theta)\right] \\
&= (2\pi)^{-d_z/2} (\nu_k^{2d_z})^{-1/2} \exp\left[-\tfrac{1}{2\nu_k^2}(y-\mu_\theta)^{\mathsf{T}}(y-\mu_\theta)\right] \\
&= (2\pi)^{-d_z/2} \nu_k^{-d_z} \exp\left[-\tfrac{1}{2\nu_k^2}(y-\mu_\theta)^{\mathsf{T}}(y-\mu_\theta)\right].
\end{aligned}$$

Taking the logarithm of each factor yields

$$\begin{aligned}
\log \pi_\theta &= \log(2\pi)^{-d_z/2} + \log \nu_k^{-d_z} + \log \exp\left[-\tfrac{1}{2\nu_k^2}(y-\mu_\theta)^{\mathsf{T}}(y-\mu_\theta)\right] \\
&= -\frac{d_z}{2}\log(2\pi) - d_z \log \nu_k - \frac{1}{2\nu_k^2}(y-\mu_\theta)^{\mathsf{T}}(y-\mu_\theta).
\end{aligned} \tag{17}$$

Let $r_\theta = y - \mu_\theta$. Because y is fixed, $\partial r_\theta / \partial \theta = 0 - J_\theta = -J_\theta$. Applying the product rule to the squared residual gives

$$\begin{aligned}\nabla_\theta(r_\theta^\mathsf{T} r_\theta) &= \left(\frac{\partial r_\theta}{\partial \theta}\right)^\mathsf{T} r_\theta + \left(\frac{\partial r_\theta}{\partial \theta}\right)^\mathsf{T} r_\theta \\ &= (-J_\theta)^\mathsf{T} r_\theta + (-J_\theta)^\mathsf{T} r_\theta = -J_\theta^\mathsf{T} r_\theta - J_\theta^\mathsf{T} r_\theta = -2J_\theta^\mathsf{T} r_\theta.\end{aligned}$$

The normalization terms have zero derivative because ν_k is fixed. Hence the score is

$$\begin{aligned}\nabla_\theta \log \pi_\theta &= 0 + 0 - \frac{1}{2\nu_k^2}\nabla_\theta(r_\theta^\mathsf{T} r_\theta) = -\frac{1}{2\nu_k^2}(-2J_\theta^\mathsf{T} r_\theta) \\ &= \frac{2}{2\nu_k^2} J_\theta^\mathsf{T} r_\theta = \frac{1}{\nu_k^2} J_\theta^\mathsf{T}(y - \mu_\theta).\end{aligned} \tag{18}$$

Next, differentiate the transition mean in Eq. (12). The prefix and scheduler interval are fixed, giving

$$\begin{aligned}J_\theta &= \frac{\partial z_{n,k}}{\partial \theta} + b_{\theta,k}\frac{\partial \Delta t_k}{\partial \theta} + \Delta t_k \frac{\partial b_{\theta,k}}{\partial \theta} \\ &= 0 + 0 + \Delta t_k \frac{\partial b_{\theta,k}}{\partial \theta} = \Delta t_k \frac{\partial b_{\theta,k}}{\partial \theta}, \qquad J_{\theta_{\text{old}}} = \Delta t_k B_{n,k}.\end{aligned}$$

Here $b_{\theta,k}$ is evaluated at $(z_{n,k}, c_n)$. Write $\epsilon = \epsilon_{n,k}^{(i)}$. The sampled residual satisfies

$$y - \mu_{\theta_{\text{old}}} = (\mu_{\theta_{\text{old}}} + \nu_k \epsilon) - \mu_{\theta_{\text{old}}} = \nu_k \epsilon.$$

Substituting the Jacobian and residual, then cancelling one factor of ν_k, gives

$$\begin{aligned}\nabla_\theta \log \pi_\theta|_{\theta_{\text{old}}} &= \frac{1}{\nu_k^2}(\Delta t_k B_{n,k})^\mathsf{T}(\nu_k \epsilon) \\ &= \frac{1}{\nu_k^2}\Delta t_k B_{n,k}^\mathsf{T} \nu_k \epsilon = \frac{\Delta t_k \nu_k}{\nu_k^2} B_{n,k}^\mathsf{T} \epsilon \\ &= \frac{\Delta t_k \nu_k}{\nu_k \nu_k} B_{n,k}^\mathsf{T} \epsilon = \frac{\Delta t_k}{\nu_k} B_{n,k}^\mathsf{T} \epsilon.\end{aligned}$$

The identity $\nabla_\theta \log \pi_\theta = (\nabla_\theta \pi_\theta)/\pi_\theta$ implies $\nabla_\theta \pi_\theta = \pi_\theta \nabla_\theta \log \pi_\theta$ by multiplication by $\pi_\theta > 0$. With the denominator fixed,

$$\begin{aligned}\nabla_\theta \rho_{k,n,i}(\theta) &= \nabla_\theta\left(\frac{\pi_\theta}{\pi_{\theta_{\text{old}}}}\right) = \frac{1}{\pi_{\theta_{\text{old}}}}\nabla_\theta \pi_\theta \\ &= \frac{1}{\pi_{\theta_{\text{old}}}}\pi_\theta \nabla_\theta \log \pi_\theta = \frac{\pi_\theta}{\pi_{\theta_{\text{old}}}}\nabla_\theta \log \pi_\theta = \rho_{k,n,i}(\theta)\nabla_\theta \log \pi_\theta.\end{aligned}$$

At the behavior parameters, $\rho_{k,n,i}(\theta_{\text{old}}) = \pi_{\theta_{\text{old}}}/\pi_{\theta_{\text{old}}} = 1$. Since $\nabla_\theta A_{n,i} = 0$, the objective gradient is

$$\begin{aligned}g_{k,n,i} &= \nabla_\theta(\rho_{k,n,i}(\theta)A_{n,i})|_{\theta_{\text{old}}} \\ &= [A_{n,i}\nabla_\theta \rho_{k,n,i}(\theta) + \rho_{k,n,i}(\theta)\nabla_\theta A_{n,i}]|_{\theta_{\text{old}}} \\ &= A_{n,i}\,\nabla_\theta \rho_{k,n,i}(\theta)|_{\theta_{\text{old}}} + 0 = A_{n,i} \cdot 1 \cdot \nabla_\theta \log \pi_\theta|_{\theta_{\text{old}}} \\ &= A_{n,i}\frac{\Delta t_k}{\nu_k} B_{n,k}^\mathsf{T} \epsilon = \frac{\Delta t_k}{\nu_k} A_{n,i} B_{n,k}^\mathsf{T} \epsilon.\end{aligned}$$

Finally, w_k is fixed with respect to θ, so multiplying the objective by w_k multiplies its gradient by the same factor. Explicitly,

$$\begin{aligned}w_k g_{k,n,i} &= \left(w_k^{\text{noise}}\frac{\Delta t_{\max}}{\Delta t_k}\right)\left(\frac{\Delta t_k}{\nu_k} A_{n,i} B_{n,k}^\mathsf{T} \epsilon\right) \\ &= \frac{w_k^{\text{noise}} \Delta t_{\max} \Delta t_k}{\Delta t_k \nu_k} A_{n,i} B_{n,k}^\mathsf{T} \epsilon = \frac{w_k^{\text{noise}} \Delta t_{\max}}{\nu_k}\frac{\Delta t_k}{\Delta t_k} A_{n,i} B_{n,k}^\mathsf{T} \epsilon \\ &= \frac{w_k^{\text{noise}} \Delta t_{\max}}{\nu_k} A_{n,i} B_{n,k}^\mathsf{T} \epsilon = \Delta t_{\max}\frac{w_k^{\text{noise}}}{\nu_k} A_{n,i} B_{n,k}^\mathsf{T} \epsilon.\end{aligned}$$

With the definition in Eq. (15), the remaining noise factor simplifies as

$$\frac{w_k^{\text{noise}}}{\nu_k} = \frac{\nu_k/\bar{\nu}}{\nu_k} = \frac{\nu_k}{\bar{\nu}\nu_k} = \frac{1}{\bar{\nu}}, \qquad w_k g_{k,n,i} = \frac{\Delta t_{\max}}{\bar{\nu}} A_{n,i} B_{n,k}^{\mathsf{T}} \epsilon_{n,k}^{(i)}.$$

□

Eq. (14) contains an explicit factor Δt_k from $\partial \mu_{\theta,k}/\partial\theta = \Delta t_k\, \partial b_{\theta,k}/\partial\theta$. Multiplication by $\Delta t_{\max}/\Delta t_k$ cancels this factor, as shown in Eq. (16). With the normalized noise weight, the explicit scalar coefficient becomes $\Delta t_{\max}/\bar{\nu}$, which is independent of k. The advantage, drift Jacobian, and sampled noise still vary across transitions, so this does not imply equal gradient norms across stages.

A.4 Variance-Efficient Timestep Allocation

We consider the allocation of a fixed sampling budget to estimate the mean reward averaged uniformly over conditions and stages. Under independent sampling and equal cost per branch, we derive the allocation that minimizes the variance of this estimator. This calculation provides a basis for the initialization in Eq. (8).

Fix the base-model parameters θ and the N conditions $(c_n, x_n^\star)$ from Sec. 3.2. For each $n \in \{1, \dots, N\}$ and $k \in \{0, \dots, K-1\}$, also fix the prefix $z_{n,k}$ and deterministic continuation $\Phi_{\theta,k+1:K}$. Let $R_{n,k}$ be the terminal reward obtained by sampling one transition from $\pi_{\theta,k}(\cdot \mid z_{n,k}, c_n)$ and completing the trajectory. Let $\eta_{n,k}$ and $\tau_{n,k}^2$ denote its conditional mean and variance. Define T_k as the root mean square of the conditional standard deviations at stage k:

$$\eta_{n,k} = \mathbb{E}[R_{n,k}], \qquad \tau_{n,k}^2 = \mathrm{Var}(R_{n,k}) < \infty, \qquad T_k = \left(\frac{1}{N}\sum_{n=1}^{N} \tau_{n,k}^2\right)^{1/2}. \tag{19}$$

Expectations and variances are taken over the sampled transition, conditional on the fixed model, conditions, and prefixes. Assume $T_k > 0$ for every stage.

Allocate $m_k \geq 1$ branches per condition to stage k, with rewards $R_{n,k}^{(i)}$ drawn independently from the distribution of $R_{n,k}$. Samples from different conditions and stages are also independent, and each branch has equal cost. For integer counts m_k, define

$$\widehat{\eta}_{n,k} = \frac{1}{m_k}\sum_{i=1}^{m_k} R_{n,k}^{(i)}, \qquad \widehat{U} = \frac{1}{NK}\sum_{n,k} \widehat{\eta}_{n,k}, \qquad U = \frac{1}{NK}\sum_{n,k} \eta_{n,k}. \tag{20}$$

Here $\sum_{n,k}$ sums over $1 \leq n \leq N$ and $0 \leq k < K$. The quantity U is the uniform average of the conditional mean rewards and $\widehat{U}$ estimates it. The budget per condition is $M = \sum_k m_k$; the total budget is NM.

Prop. A.2 (Variance-efficient allocation)**.** *Under the conditions above, $\widehat{U}$ is unbiased for U, with*

$$\mathrm{Var}(\widehat{U}) = \frac{1}{NK^2}\sum_{k=0}^{K-1} \frac{T_k^2}{m_k}. \tag{21}$$

For a fixed $M > 0$, minimizing this expression over positive real allocations m_k with $\sum_k m_k = M$ gives the unique solution

$$m_k^* = M\frac{T_k}{\sum_{j=0}^{K-1} T_j}, \qquad q_k^* = \frac{m_k^*}{M} = \frac{T_k}{\sum_{j=0}^{K-1} T_j}. \tag{22}$$

This optimization is the continuous relaxation of the integer allocation problem.

Proof. Linearity of expectation first gives the mean of each group estimator,

$$\mathbb{E}[\widehat{\eta}_{n,k}] = \mathbb{E}\left[\frac{1}{m_k}\sum_{i=1}^{m_k} R_{n,k}^{(i)}\right] = \frac{1}{m_k}\sum_{i=1}^{m_k} \mathbb{E}[R_{n,k}^{(i)}]$$
$$= \frac{1}{m_k}\sum_{i=1}^{m_k} \eta_{n,k} = \frac{m_k \eta_{n,k}}{m_k} = \eta_{n,k}.$$

Applying the same rule to the average over conditions and stages gives

$$\mathbb{E}[\widehat{U}] = \mathbb{E}\left[\frac{1}{NK}\sum_{n=1}^{N}\sum_{k=0}^{K-1}\widehat{\eta}_{n,k}\right] = \frac{1}{NK}\sum_{n=1}^{N}\sum_{k=0}^{K-1}\mathbb{E}[\widehat{\eta}_{n,k}]$$
$$= \frac{1}{NK}\sum_{n=1}^{N}\sum_{k=0}^{K-1}\eta_{n,k} = U.$$

Within a group, independent samples have zero pairwise covariance. Using the variance of a sum and the squared scaling factor,

$$\mathrm{Var}(\widehat{\eta}_{n,k}) = \frac{1}{m_k^2}\,\mathrm{Var}\left(\sum_{i=1}^{m_k} R_{n,k}^{(i)}\right)$$
$$= \frac{1}{m_k^2}\left[\sum_{i=1}^{m_k}\mathrm{Var}(R_{n,k}^{(i)}) + 2\sum_{i<j}\mathrm{Cov}(R_{n,k}^{(i)}, R_{n,k}^{(j)})\right]$$
$$= \frac{1}{m_k^2}\left[\sum_{i=1}^{m_k}\tau_{n,k}^2 + 0\right] = \frac{m_k\tau_{n,k}^2}{m_k^2} = \frac{m_k\tau_{n,k}^2}{m_k m_k} = \frac{\tau_{n,k}^2}{m_k}.$$

For distinct groups $(n,k) \neq (n',k')$, independence also gives $\mathrm{Cov}(\widehat{\eta}_{n,k}, \widehat{\eta}_{n',k'}) = 0$. Order these pairs lexicographically when writing the covariance sum below. Then

$$\mathrm{Var}(\widehat{U}) = \frac{1}{N^2K^2}\,\mathrm{Var}\left(\sum_{n,k}\widehat{\eta}_{n,k}\right)$$
$$= \frac{1}{N^2K^2}\left[\sum_{n,k}\mathrm{Var}(\widehat{\eta}_{n,k}) + 2\sum_{(n,k)<(n',k')}\mathrm{Cov}(\widehat{\eta}_{n,k}, \widehat{\eta}_{n',k'})\right]$$
$$= \frac{1}{N^2K^2}\sum_{n=1}^{N}\sum_{k=0}^{K-1}\mathrm{Var}(\widehat{\eta}_{n,k}) = \frac{1}{N^2K^2}\sum_{n=1}^{N}\sum_{k=0}^{K-1}\frac{\tau_{n,k}^2}{m_k}$$
$$= \frac{1}{N^2K^2}\sum_{k=0}^{K-1}\frac{1}{m_k}\sum_{n=1}^{N}\tau_{n,k}^2.$$

The definition of T_k implies, successively,

$$T_k^2 = \frac{1}{N}\sum_{n=1}^{N}\tau_{n,k}^2, \qquad NT_k^2 = \sum_{n=1}^{N}\tau_{n,k}^2.$$

Substitution and cancellation of N now give

$$\mathrm{Var}(\widehat{U}) = \frac{1}{N^2K^2}\sum_{k=0}^{K-1}\frac{NT_k^2}{m_k} = \frac{N}{N^2K^2}\sum_{k=0}^{K-1}\frac{T_k^2}{m_k}$$
$$= \frac{N}{NNK^2}\sum_{k=0}^{K-1}\frac{T_k^2}{m_k} = \frac{1}{NK^2}\sum_{k=0}^{K-1}\frac{T_k^2}{m_k}.$$

The positive constant $1/(NK^2)$ does not affect the minimizer. Set $F(m) = \sum_k T_k^2/m_k$ and $S_T = \sum_k T_k > 0$. For positive real m_k, apply Cauchy–Schwarz to $u_k = T_k/\sqrt{m_k}$ and $v_k = \sqrt{m_k}$. Their product is $u_k v_k = T_k\sqrt{m_k}/\sqrt{m_k} = T_k$, so

$$S_T^2 = \left(\sum_k u_k v_k\right)^2 \leq \left(\sum_k u_k^2\right)\left(\sum_k v_k^2\right)$$
$$= \left(\sum_k \frac{T_k^2}{m_k}\right)\left(\sum_k m_k\right) = F(m)M.$$

Dividing by the positive budget M yields

$$\frac{S_T^2}{M} \le \frac{F(m)M}{M} = F(m).$$

Equality holds exactly when $u_k = av_k$ for a common $a > 0$. For each stage, multiplying by $\sqrt{m_k}$ and then dividing by a gives

$$\frac{T_k}{\sqrt{m_k}} = a\sqrt{m_k}, \qquad \frac{T_k}{\sqrt{m_k}}\sqrt{m_k} = a\sqrt{m_k}\sqrt{m_k},$$
$$T_k = am_k, \qquad \frac{T_k}{a} = \frac{am_k}{a}, \qquad m_k = \frac{T_k}{a}.$$

Imposing the budget determines a,

$$M = \sum_k m_k = \sum_k \frac{T_k}{a} = \frac{S_T}{a},$$
$$aM = a\frac{S_T}{a} = S_T, \qquad \frac{aM}{M} = \frac{S_T}{M}, \qquad a = \frac{S_T}{M}.$$

Substituting this value back into m_k and dividing by M gives

$$m_k^* = \frac{T_k}{S_T/M} = T_k\frac{M}{S_T} = \frac{MT_k}{S_T},$$
$$q_k^* = \frac{m_k^*}{M} = \frac{MT_k/S_T}{M} = \frac{MT_k}{MS_T} = \frac{T_k}{S_T}.$$

All these allocations are positive. Their sum and objective value are

$$\sum_k m_k^* = \sum_k \frac{MT_k}{S_T} = \frac{M}{S_T}\sum_k T_k = \frac{MS_T}{S_T} = M,$$
$$F(m^*) = \sum_k \frac{T_k^2}{MT_k/S_T} = \sum_k \frac{S_T T_k^2}{MT_k} = \sum_k \frac{S_T T_k}{M}$$
$$= \frac{S_T}{M}\sum_k T_k = \frac{S_T^2}{M}.$$

Thus the allocation is feasible and attains the lower bound. Every minimizer must attain equality in Cauchy–Schwarz. Since $T_k > 0$ for all k, that equality and the budget uniquely fix a and every m_k. □

Eq. (22) gives the allocation ratio

$$\frac{m_k^*}{m_j^*} = \frac{MT_k/S_T}{MT_j/S_T} = \frac{MT_k}{S_T}\frac{S_T}{MT_j} = \frac{MT_kS_T}{S_TMT_j} = \frac{T_k}{T_j}.$$

Here $S_T = \sum_j T_j$, as defined in the proof. Thus the optimal allocation ratio equals the ratio of stage dispersions. The initialization in Eq. (8) uses $S(k) = N^{-1}\sum_n s_{n,k}$, where $s_{n,k}$ is the sample standard deviation from G diagnostic branches. Its population counterpart is $\bar{\tau}_k = N^{-1}\sum_n \tau_{n,k}$, the arithmetic mean of the conditional standard deviations. To relate this to the root mean square T_k, first note that $\sum_n \tau_{n,k} = N\bar{\tau}_k$. Expanding the squared deviations gives

$$\begin{aligned}\frac{1}{N}\sum_{n=1}^{N}(\tau_{n,k} - \bar{\tau}_k)^2 &= \frac{1}{N}\sum_{n=1}^{N}(\tau_{n,k}^2 - 2\tau_{n,k}\bar{\tau}_k + \bar{\tau}_k^2)\\ &= \frac{1}{N}\sum_n \tau_{n,k}^2 - \frac{2\bar{\tau}_k}{N}\sum_n \tau_{n,k} + \frac{1}{N}\sum_n \bar{\tau}_k^2\\ &= T_k^2 - \frac{2\bar{\tau}_k}{N}(N\bar{\tau}_k) + \frac{N\bar{\tau}_k^2}{N} = T_k^2 - \frac{2N\bar{\tau}_k^2}{N} + \bar{\tau}_k^2\\ &= T_k^2 - 2\bar{\tau}_k^2 + \bar{\tau}_k^2 = T_k^2 - \bar{\tau}_k^2.\end{aligned} \tag{23}$$

Thus $T_k = \bar{\tau}_k$ if the conditional standard deviations are equal across conditions at stage k. In general, $S(k)$ is an empirical proxy used to initialize the timestep distribution; Prop. A.2 establishes optimality for T_k in estimating U, not for $S(k)$ in policy optimization.