Understanding Sampler Stochasticity in Training Diffusion Models for RLHF

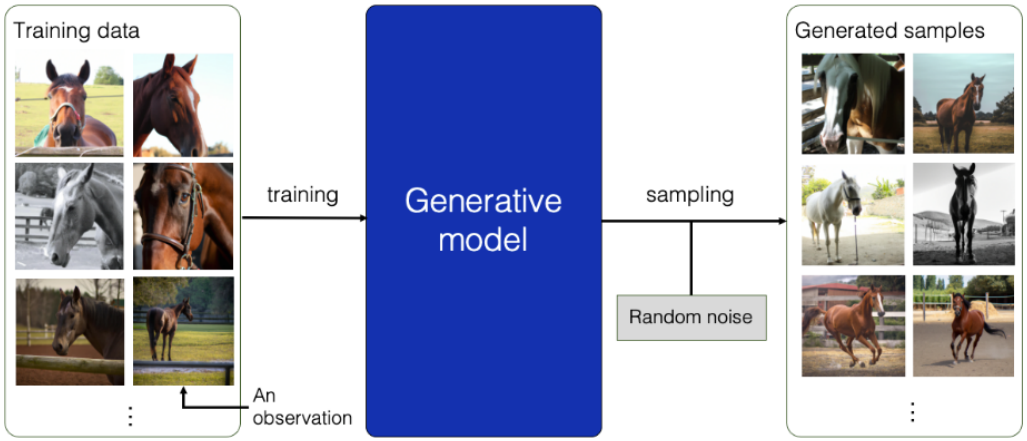
Columbia University, IEOR Department

Wenpin Tang

Generative modeling

Generative modeling has advanced tremendously:

- Generative adversary networks (GANs),
- Variational autoencoders (VAEs),
- and more recently, LLMs $\approx$ Autoregressive models.



Applications: Chatbot (text), multimodal objects (image, audio,...), data creation (healthcare, finance), etc.

(Pretrained) Diffusion Model

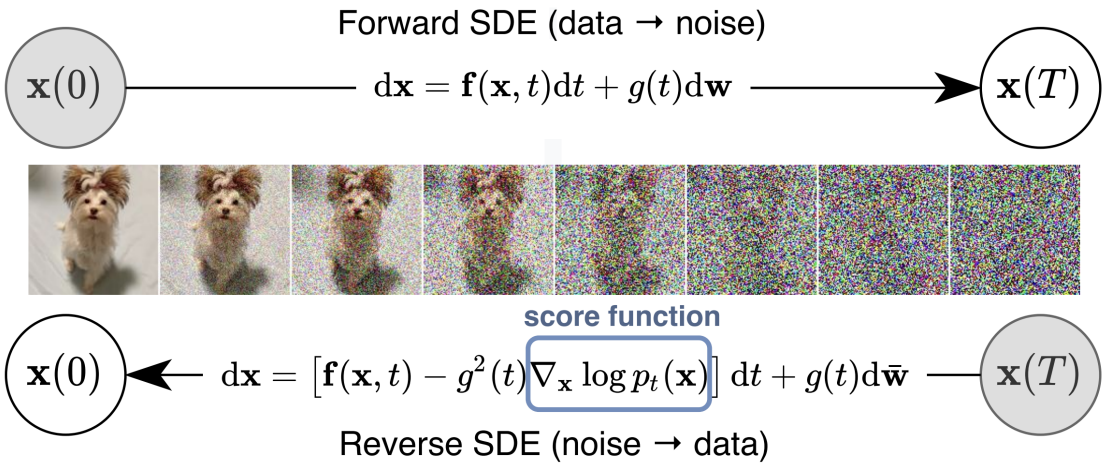
Question: How to create a good data?

A simple (but not good) approach: suppose you have a set of data, and you want to create a *similar* but *different* thing.

- Pick one sample from the empirical data distribution.
- Perturb it.

If the perturbation is (too) small, then you don't create anything.
If the perturbation is large, then you may get "garbage".

Key: perturb "smartly".



Two popular examples

- Variance Exploding (VE) Model

$$dX_t = \sqrt{\frac{d\sigma^2(t)}{dt}} dB_t = \sqrt{2\sigma(t)} dB_t,$$

with the corresponding discretized version

$$x_i = x_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2} z_{i-1}, \quad z_i \sim^{\text{i.i.d}} \mathcal{N}(0, I)$$

- Variance Preserving (VP) Model

$$dX_t = -\frac{1}{2}\beta(t) X_t dt + \sqrt{\beta(t)},$$

with the corresponding discretized version

$$x_i = \sqrt{1-\beta_i} x_{i-1} + \sqrt{\beta_i} z_i, \quad z_i \sim^{\text{i.i.d}} \mathcal{N}(0, I)$$

Post-training

Pretraining: large models trained on a large dataset (e.g., Stable Diffusion, Nano Banana)

Most relevantly, we need to *customize* to specific goals, e.g.,

- Improve the aesthetic quality of an image;
- Incorporate some functional properties to a data set.

Idea: Reinforcement Learning from Human Feedback (RLHF)

Question: How to balance the exploration in diffusion models?
Diffusion sampler with *stochasticity* η:

$$dY_t = \left(-f(T-t, Y_t) + \frac{1+\eta^2}{2} g^2(T-t) s_\theta(T-t, Y_t)\right) dt + \eta\, g(T-t) dB_t, \quad Y_0 \sim p_{\text{noise}}(\cdot),$$

because the distribution of Y_T is preserved, $\forall \eta > 0$.

Stochasticity in post-training

Two (contradictory) practices:

- RL for post-training: require exploration $\rightarrow$ large η;
- Fast inference: ODE samplers $\rightarrow \eta = 0$.

Common/current practices: Use a medium stochasticity ($\eta = 0.7 \sim 1$) in generating samples for training, and the ODE sampler ($\eta = 0$) for inference.

Two questions:

- Can we have good sampling quality when inference uses a different noise level than the one used in RLHF training?
  $\rightarrow$ Reward gap/mismatch (Reward: ImageReward, z-score, etc);
- All the previous works use $\eta \in [0, 1]$, interpolating DDIM ($\eta = 0$) and DDPM ($\eta = 1$). What's wrong with $\eta > 1$?

We consider different sub-sequences τ of $[1, \ldots, T]$ and different variance hyperparameters σ indexed by elements of τ. To simplify comparisons, we consider σ with the form:

$$\sigma_{\tau_i}(\eta) = \eta \sqrt{(1 - \alpha_{\tau_{i-1}})/(1 - \alpha_{\tau_i})} \sqrt{1 - \alpha_{\tau_i}/\alpha_{\tau_{i-1}}}, \tag{16}$$

where $\eta \in \mathbb{R}_{\geq 0}$ is a hyperparameter that we can directly control. This includes an original DDPM generative process when $\eta = 1$ and DDIM when $\eta = 0$. We also consider DDPM where the random noise has a larger standard deviation than $\sigma(1)$, which we denote as $\hat{\sigma}$: $\hat{\sigma}_{\tau_i} = \sqrt{1 - \alpha_{\tau_i}/\alpha_{\tau_{i-1}}}$. This is used by the implementation in Ho et al. (2020) **only to obtain the CIFAR10 samples**, but not samples of the other datasets. We include more details in Appendix D.

Back-of-the-envelop (I)

VE (Variance Exploding):

$$dX_t = \sqrt{2t}\, dB_t, \quad \text{with } p_{\text{data}}(\cdot) = \mathcal{N}(0,1).$$

- Calculate the *exact* backward dynamics:

$$\mathrm{f}(x,t) = 0, \quad \mathrm{g}(x,t) = \sqrt{2t}, \quad s(x,t) := \nabla_X \log p_t(x) = -\frac{x}{t^2+1}.$$

- Pretrained Model:

$$dY_t^{REF} = -\frac{(1+\eta^2)(T-t)}{1+(T-t)^2} Y_t^{REF} dt + \eta\sqrt{2(T-t)} dB_t.$$

- Post-training as a simple control (the *worst* post-training):

$$dY_t^{SDE} = -\frac{(1+\eta^2)(T-t)}{1+(T-t)^2}(Y_t^{SDE} + \theta_\eta^*)dt + \eta\sqrt{2(T-t)} dB_t,$$

$$dY_t^{ODE} = -\frac{T-t}{1+(T-t)^2}(Y_t^{ODE} + \theta_\eta^*)dt.$$

Back-of-the-envelop (II)

Training: Write $\{Y_t^{SDE}\} = \{Y_t^{\theta}\}$, solve:

$$\max_{\theta} \mathbb{E}[r(Y_T^{\theta})] - \beta KL(Y_T^{\theta}, Y_T^{REF}).$$

Here we take $r(y) = -(y-1)^2 \rightarrow$ preference shift to 1.

Definition

1. Improvement of fine-tuning as the reward difference between $\{Y_t^{ODE}\}$ and $\{Y_t^{REF}\}$, $I_\eta := \mathbb{E}[r(Y_T^{ODE})] - \mathbb{E}[r(Y_T^{REF})]$.
2. Reward gap as the reward difference between $\{Y_t^{ODE}\}$ and $\{Y_t^{SDE}\}$, $\Delta_\eta := \mathbb{E}[r(Y_T^{ODE})] - \mathbb{E}[r(Y_T^{SDE})]$.

Back-of-the-envelop (III)

Theorem

Consider the VE model, with the reward function $r(y) = -(y-1)^2$. *For* $\eta > 0$, *we have* $\theta^*_\eta = -\left[(1+\frac{\beta}{2})\left(1-(1+T^2)^{-\frac{1+\eta^2}{2}}\right)\right]^{-1}$. *Moreover,*

$$\Delta_\eta \leq \frac{1}{2T} + o\left(\frac{1}{T}\right) \quad \textit{and} \quad \mathcal{I}_\eta \geq 1 - \frac{1}{2T} + o\left(\frac{1}{T}\right).$$

Remarks:

- The bounds are independent of η, and $\Delta_\eta \to 0$ as $T \to \infty$.
- Also work for the VP case, with $\Delta_\eta \leq \frac{e^{-T^2}}{2} + o(e^{-T^2})$.
- Also works in mixture Gaussian setting, with quadratic rewards.

gDDIM (I)

Recall the backward sampler:

$$dY_t = \left(-f + \frac{1+\eta^2}{2} g^2 s_\theta\right) dt + \eta\, g\, dB_t.$$

Use the parametrization $f(t,x) = \frac{1}{2}\frac{d\log\alpha_t}{dt}$ and $g(t) = \sqrt{-\frac{d\log\alpha_t}{dt}}$, where α_t is decreasing with $\alpha_0 = 1$ and $\alpha_T = 0$.

Fact: The "correct" discretization is:

$$x_{t-\Delta} = \sqrt{\frac{\alpha_{t-\Delta}}{\alpha_t}} x_t + \left(\sqrt{\frac{\alpha_{t-\Delta}}{\alpha_t}}(1-\alpha_t) - \sqrt{(1-\alpha_{t-\Delta}-\sigma_t^2)(1-\alpha_t)}\right) s_\theta + \sigma_t(\eta)\mathcal{N}(0, I),$$

where $\{x_t\}_{t=0}^T$ is the discretized sequence of backward diffusion Y_t, and

$$\sigma_t(\eta) = (1-\alpha_{t-\Delta t})\left(1 - \left(\frac{1-\alpha_{t-\Delta t}}{1-\alpha_t}\right)^{\eta^2}\left(\frac{\alpha_t}{\alpha_{t-\Delta t}}\right)^{\eta^2}\right).$$

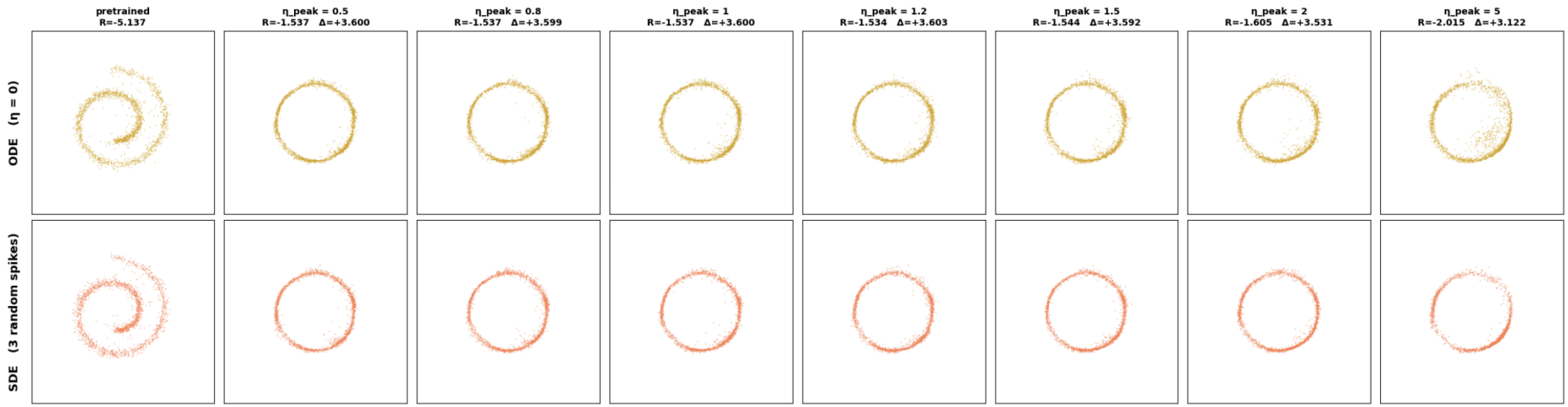
gDDIM (II)

Idea: If p_0 is Dirac (or Gaussian), we can recover the score $\nabla \log p_t(u)$ from any evaluation of $\nabla \log p_s(u(s))$, where $u(s)$ is any solution to the backward SDE given $u(t) = u$ (Tweedie) + Exponential integrator.

Theorem

Assume that $|s_\theta| \leq A$. *Under the gDDIM scheme,*

$$W_2(x_T^{ODE}, x_T^{SDE}) \leq C(T, A)\left(1 - e^{-C\eta^2}\right).$$



Experiments

Base models: Stable Diffusion v1.5 and FLUX v1

RLHF algorithms:

- Denoising Diffusion Policy Optimization (DDPO): Directly optimize reward function by formulate the denoising steps as a Markov Decision Process.
- Mixed Group Relative Policy Optimization (MixGRPO): Use a mixture of SDE-ODE sampling and optimize based on group-relative advantages.

Reward metrics: aesthetic score, ImageReward, PickScore, HPSv2.

DDPO

We formulate the denoising steps $\{x_T, x_{T-1}, \cdots, x_0\}$ as an MDP:

$$s_t := (\mathrm{c}, t, x_t),\ \pi(a_t|s_t) := p_\theta(x_{t-1}|x_t, \mathrm{c}),\ \mathbb{P}(s_{t+1}|s_t, a_t) := \left(\delta_{\mathrm{c}}, \delta_{t-1}, \delta_{x_{t-1}}\right),$$

$$a_t := x_{t-1},\ \rho_0(s_0) := \left(p(\mathrm{c}), \delta_T, \mathcal{N}(0, I)\right),\ R(s_t, a_t) := \begin{cases} r(x_0, \mathrm{c}) & (t = 0), \\ 0 & (t > 0). \end{cases}$$

Here

- p_θ is the probability from x_t to x_{t-1} given prompt c;
- θ-parametrized score function s_θ;
- $\delta_\bullet$ is the Dirac function that distributes the (non-zero) mass at $\bullet$ only;
- R is the reward function that is equal to the denoised image quality $r(x_0, c)$ for each trajectory.

DDPO: High Stochasticity Benefits Moderate Time Steps

	η	ImageReward	PickScore	HPS_v2	Aesthetic
Base	–	0.320	20.69	0.253	5.375
ImageReward	1.0	0.837 (0.918)	20.89 (20.94)	**0.268** (**0.271**)	5.638 (5.720)
	1.2	**0.915** (**1.031**)	**20.95** (**20.99**)	**0.268** (0.289)	**5.697** (**5.803**)
	1.5	0.771 (0.907)	20.90 (20.95)	0.266 (0.269)	5.605 (5.735)
PickScore	1.0	0.587 (**0.729**)	21.11 (21.28)	**0.265** (**0.267**)	5.560 (5.678)
	1.2	**0.594** (0.692)	**21.17** (21.33)	0.264 (0.264)	5.568 (5.692)
	1.5	0.566 (0.701)	21.10 (**21.36**)	0.264 (0.262)	**5.599** (**5.769**)

Table 1: Performance for ODE (SDE) samplers under DDPO fine-tuning. Bold numbers indicate the highest evaluations among all stochasticity, demonstrating $\eta = 1.2$ in general performs the best.

DDPO: Decreasing Reward Gap with Quality Improvement

	$T = 0$	**200**	**400**	**600**	**800**	**1000**	**1200**	**1400**	**(1476)**
IR	0.160	0.119	0.102	0.028	0.057	0.027	**0.006**	0.020	(0.564)
HPS	0.0047	0.0032	0.0032	0.0018	0.0027	0.0015	**-0.0028**	0.0011	(0.0308)
Aes	0.162	0.113	0.078	0.077	0.072	0.017	0.030	**-0.010**	(0.685)
PS	0.115	0.146	0.167	0.118	0.178	0.106	0.129	**0.090**	(0.907)
SDE	20.823	21.310	21.647	21.901	22.068	22.265	22.306	**22.417**	(18.920)
ODE	20.730	21.165	21.500	21.783	21.883	22.172	22.195	**22.318**	(17.044)

Table 2: PickScore training until reward collapses. Smallest reward gaps and best sampler performances locate at the large training steps $T = 1200, 1400$.

DDPO: Richer Prompt Contents Reduce Reward Gap

	Animal Prompts			Comprehensive Prompts		
	$T = 0$	100	200	$T = 0$	100	200
Mean	**0.382 (0.539)**	0.668 (0.805)	0.915 (1.031)	0.347 (0.470)	**0.763 (0.872)**	**1.086 (1.174)**
Std	0.814 (0.799)	0.775 (0.721)	0.709 (0.652)	**1.030 (0.997)**	**0.909 (0.868)**	**0.795 (0.727)**
Gap	**0.146**	**0.138**	**0.116**	0.115	0.110	0.106

Table 3: Performance Comparison between prompts of different complexity. More complicated prompts yields faster fine-tuning improvements, larger in-group variances, and smaller SDE-ODE reward gaps.

DDPO: Reward Gap Visualization

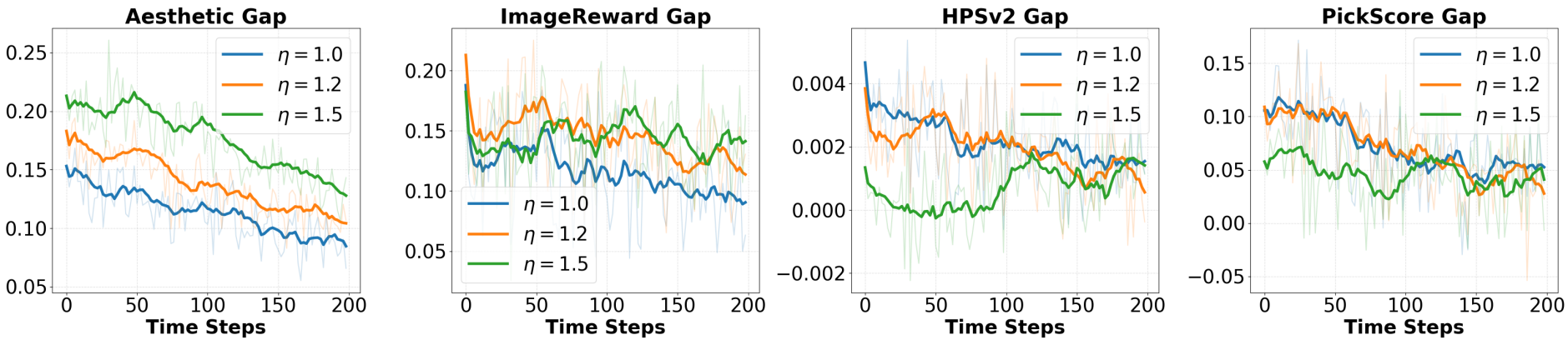


Figure 1: Evolution of reward gap during DDPO training under PickScore fine-tuning with stochasticity $\eta \in \{1, 1.2, 1.5\}$.

The gap for multiple results decreases steadily as training progresses, indicating that image quality improves for both samplers and that ODE inference remains competitive

DDPO: Image Visualization

DDPO Images under PickScore

Figure 2: Sampling schemes (from left to right): (i) SDE with $\eta = 0.75$, (ii) ODE with $\eta = 0.75$; (iii) SDE with $\eta = 1.5$, (iv) ODE with $\eta = 1.5$. Prompts (from top to bottom): “baboon", “white wolf, Arctic wolf", “clumber, clumber spaniel".

GRPO

GRPO uses a group of sampled outputs with the group normalization/average per prompt, to compute the *advantages*:

$$A_i := \frac{r(x_0^i, \mathrm{c}) - \frac{1}{G}\sum_{k=1}^{G} r(x_0^k, \mathrm{c})}{\mathrm{std}\left(\{r(x_0^k, \mathrm{c})\}_{k=1}^{G}\right)},$$

where G is the group size; $\mathrm{std}\left(\{r(x_0^k, \mathrm{c})\}_{k=1}^{G}\right)$ denotes the standard deviation of the group rewards. The objective of GRPO is to maximize:

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{T}\sum_{t=1}^{T}\min\left(\rho_t^i(\theta)A_i, \mathrm{clip}(\rho_t^i(\theta), 1-\epsilon, 1+\epsilon)A_i\right)\right],$$

where $\rho_t^i(\theta) = \frac{p_\theta(x_{t-1}^i | x_t^i, \mathrm{c})}{p_{\theta_{\mathrm{old}}}(x_{t-1}^i | x_t^i, \mathrm{c})}$ is the likelihood ratio; $\{x_t^i\}$ is the i-th sample trajectory in the group; and the expectation is with respect to $\mathrm{c} \sim p(\mathrm{c})$ and $\{x_t^i\} \sim \pi_{\theta_{\mathrm{old}}}(\cdot|\mathrm{c})$.

MixGRPO

MixGRPO uses GRPO-style objective and a combination mix ODE and SDE for sampling. The specific form is as follows:

$$\mathrm{d}\mathrm{x}_t = \begin{cases} \left[f(\mathrm{x}_t, t) - g^2(t)\nabla_{\mathrm{x}_t} \log q_t(\mathrm{x}_t)\right] \mathrm{d}t + g(t)\mathrm{dw}, & \text{if } t \in S, \\ \left[f(\mathrm{x}_t, t) - \frac{1}{2} g^2(t)\nabla_{\mathrm{x}_t} \log q_t(\mathrm{x}_t)\right] \mathrm{d}t, & \text{otherwise.} \end{cases}$$

With Euler-Maruyama discretization for SDE and Euler discretization for ODE, the final denoising process follows:

$$\mathrm{x}_{t+\Delta t} = \begin{cases} \mathrm{x}_t + \left[\mathrm{v}_\theta(\mathrm{x}_t, t) + \frac{\sigma_t^2(\mathrm{x}_t + (1-t)\mathrm{v}_\theta(\mathrm{x}_t, t))}{2t}\right] \Delta t + \sigma_t \sqrt{\Delta t}\epsilon, & \text{if } t \in S \\ \mathrm{x}_t + \mathrm{v}_\theta(\mathrm{x}_t, t)\Delta t, & \text{otherwise} \end{cases}$$

MixGRPO: Bounded Reward Gap and Performance Improvement

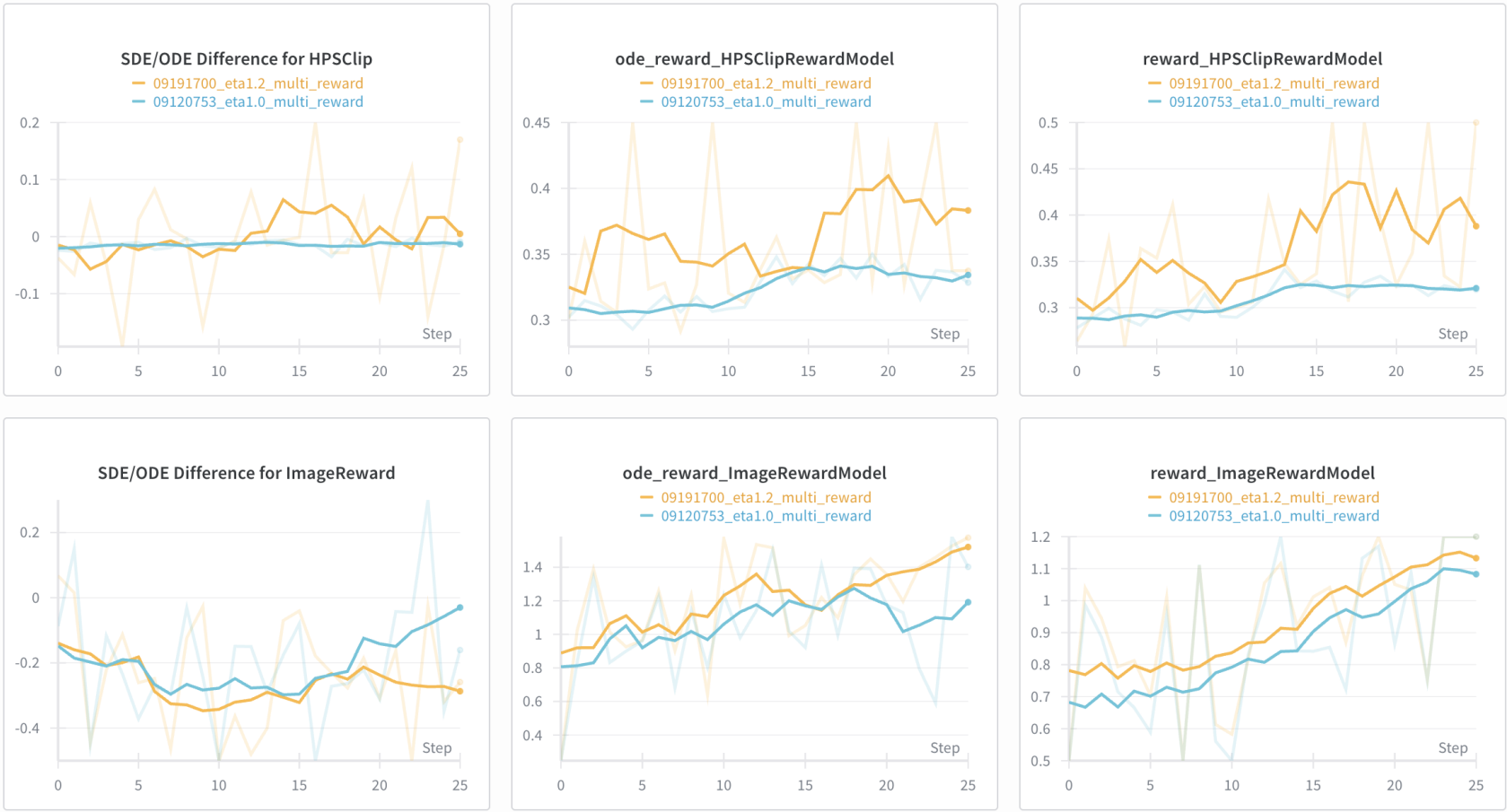


Figure 3: Bounded reward gap (left column) and performance improvement (middle, right column) for MixGRPO

MixGRPO: Image Visualization

Figure 4: Comparison of ODE image generation.
Prompt (leftmost): "A steampunk pocketwatch owl is trapped inside a glass jar buried in sand, surrounded by an hourglass and swirling mist."

Questions?

Next time, you will probably hear from me:

- *Tweedie's formula beyond Gaussian*: applications to diffusion models and empirical Bayes, with Nizar Touzi, Zikun Zhang and Xun Yu Zhou.
- *Post-training diffusion models via deterministic adaptive adjoint matching*, with Zhengyi Guo and Jiayuan Sheng.
- *Post-training diffusion LLM via score as action*, with Zikun Zhang and Jiayuan Sheng.
- *Embodied AI from control perspectives*: inner reflection (LLM) and outer stimulus (RL), with xxx.