PPO Fine-Tuning of Diffusion Models: Provable Convergence across Interpolated Trajectories

Ruinan Jin[1], Mingyi Wang[2], Yuchen Liang[1], Shaofeng Zou[2], Ness Shroff[1], **Yingbin Liang**[1]

The Ohio State University[1]

Arizona State University[2]

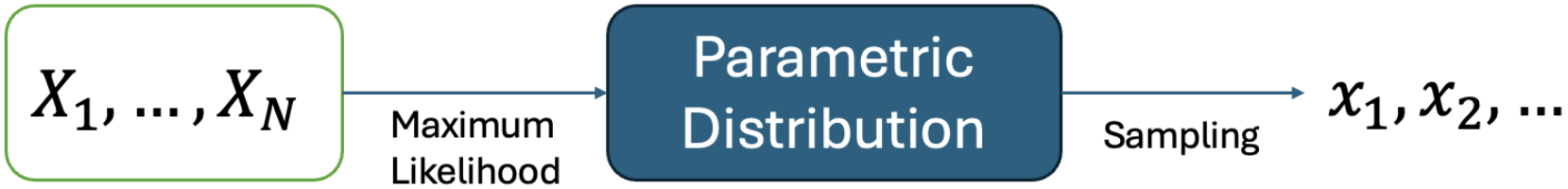
April 24, 2026

Generative Modeling

Goal: Learn underlying distribution of data, so that generating new samples that look like training data

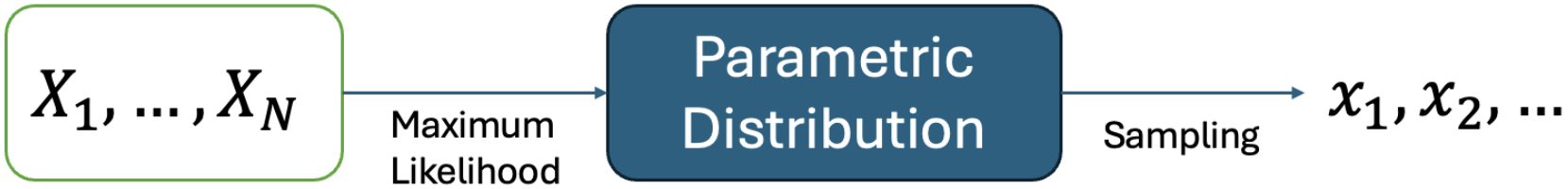
Generative Modeling

Goal: Learn underlying distribution of data, so that generating new samples that look like training data



Generative Modeling

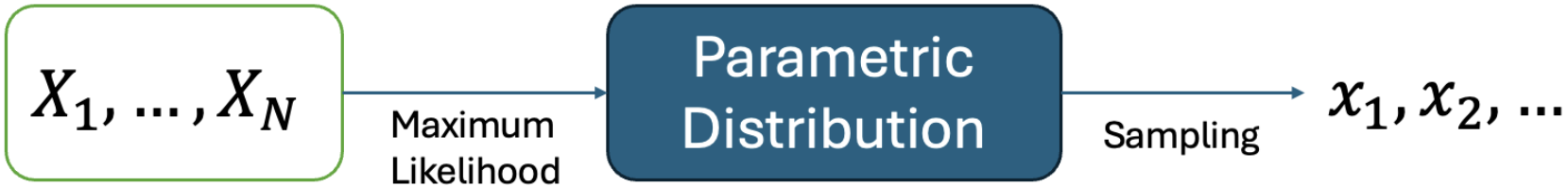
Goal: Learn underlying distribution of data, so that generating new samples that look like training data



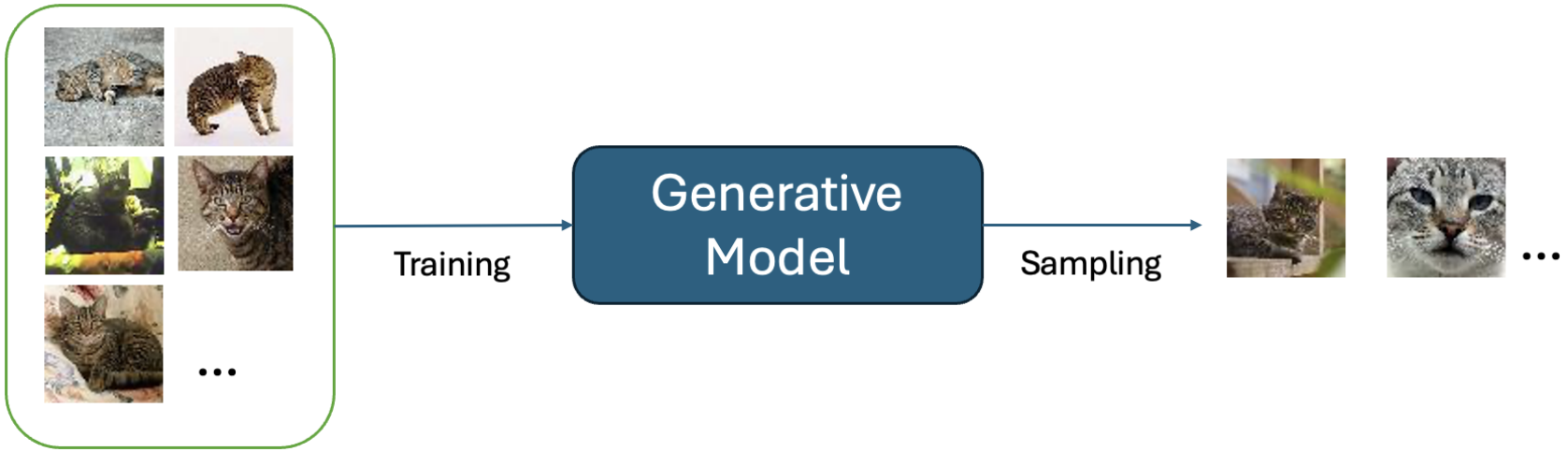
Challenges: (i) data distribution is high-dimensional and complex; (ii) capture fine-grained structure; (iii) generate diverse yet realistic samples

Generative Modeling

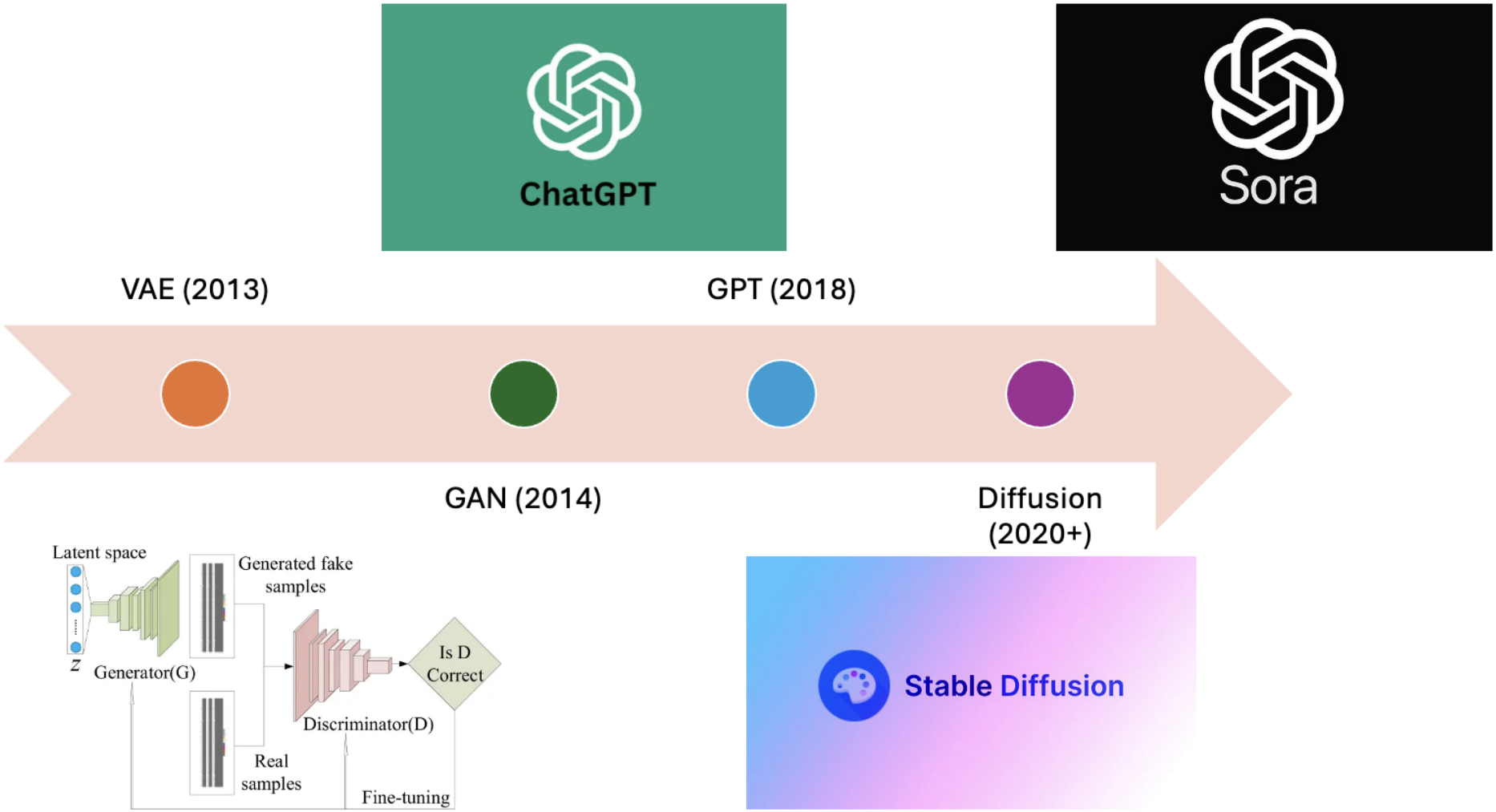
Goal: Learn underlying distribution of data, so that generating new samples that look like training data



Challenges: (i) data distribution is high-dimensional and complex; (ii) capture fine-grained structure; (iii) generate diverse yet realistic samples

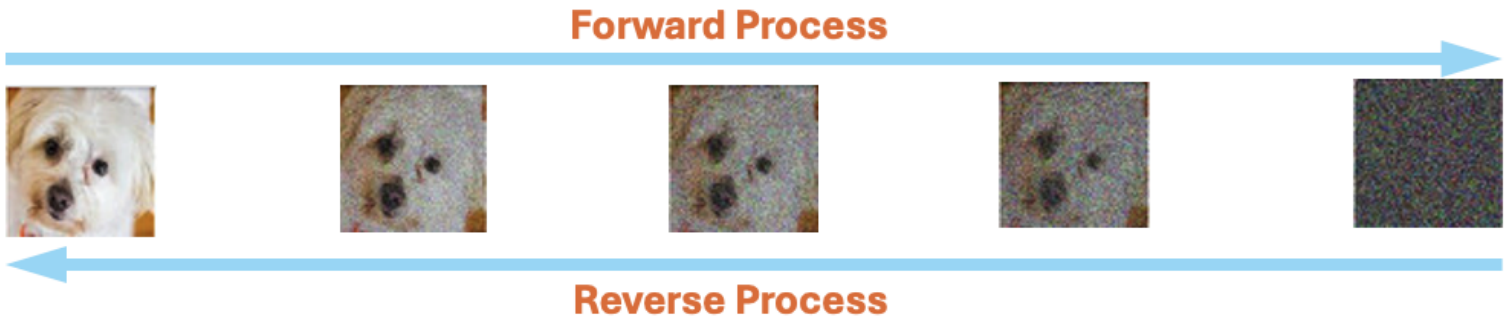


Timeline of Generative AI



Background: Diffusion Generative Models

Diffusion models have become a dominant paradigm for high-fidelity generative modeling (Ho et al. 2020; Song et al. 2021).

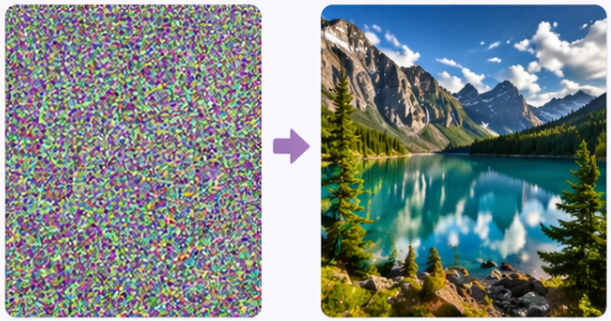


Applications of Diffusion Models

Image Generation

Create realistic and high-quality images from random noise.

Input (random noise) → Generated image



Text Generation

Generate coherent and context-aware text.

Input (random noise)

the mean tree help
quick brown see new
idea climate paper sun
future science model
data learn compute
language diffusion ...

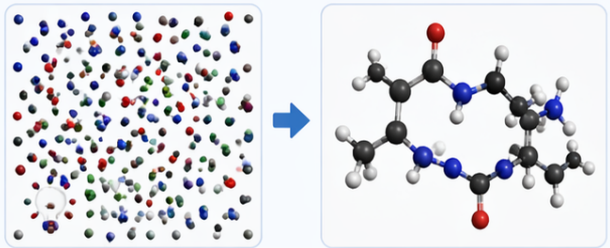
Generated text

Diffusion models are a class of **generative models** that learn to reverse a gradual noising process. They have shown remarkable performance in a **wide range of tasks**, including image synthesis, text generation, and structured data modeling.

Molecule Generation

Design novel molecules with desired properties.

Input (random noise) → Generated molecule

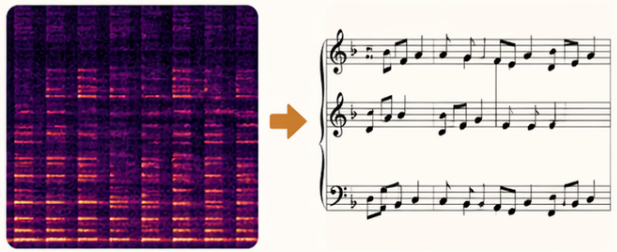


Example property: **High predicted binding affinity** to a target protein.

Music Generation

Compose realistic and diverse music.

Input (random noise) → Generated music (melody)



0:00 / 0:30

Why Post-Training of Diffusion Models?

- **Pre-trained diffusion models** generate realistic samples, but many downstream goals are not captured by likelihood pretraining alone.
- In practice, we often want outputs that optimize task-specific rewards, for example:
  - human preference
  - safety and harmlessness
  - scientific properties such as binding affinity or stability

Generating audience-aligned text

Task: Explain a solar eclipse

Pre-trained model: general generation

A solar eclipse occurs when the Moon passes between the Earth and the Sun, obscuring the solar photosphere for observers located within the lunar umbra or penumbra. The apparent coverage depends on the Sun–Moon–Earth geometry and the observer's position.

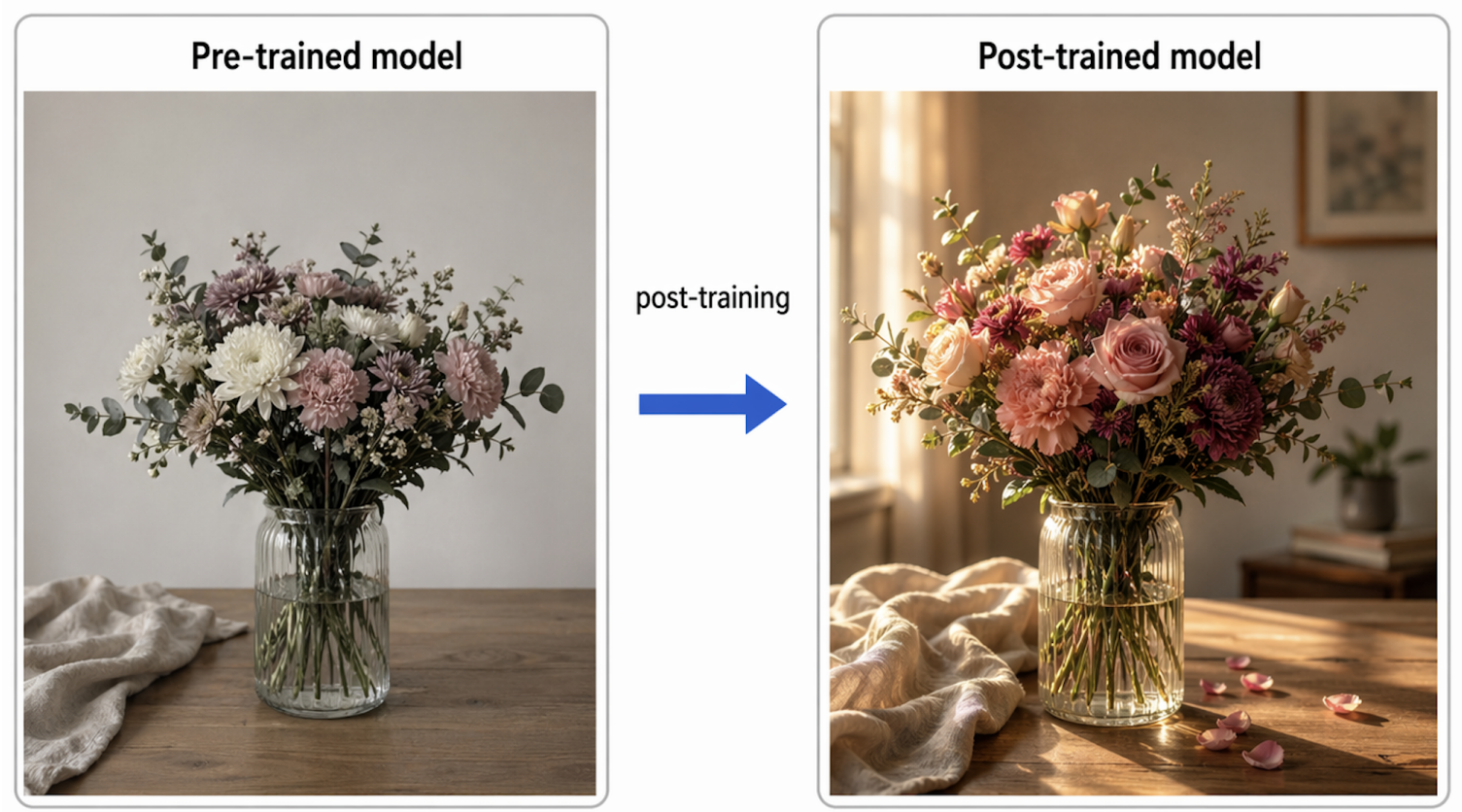
Downstream need: explain in an easy-to-understand manner

Post-trained model: provide child-friendly explanation

A solar eclipse happens when the Moon moves in front of the Sun and blocks its light for a little while. The Moon likely makes a shadow on Earth. That is why the Sun can look darker or even disappear for a short time.

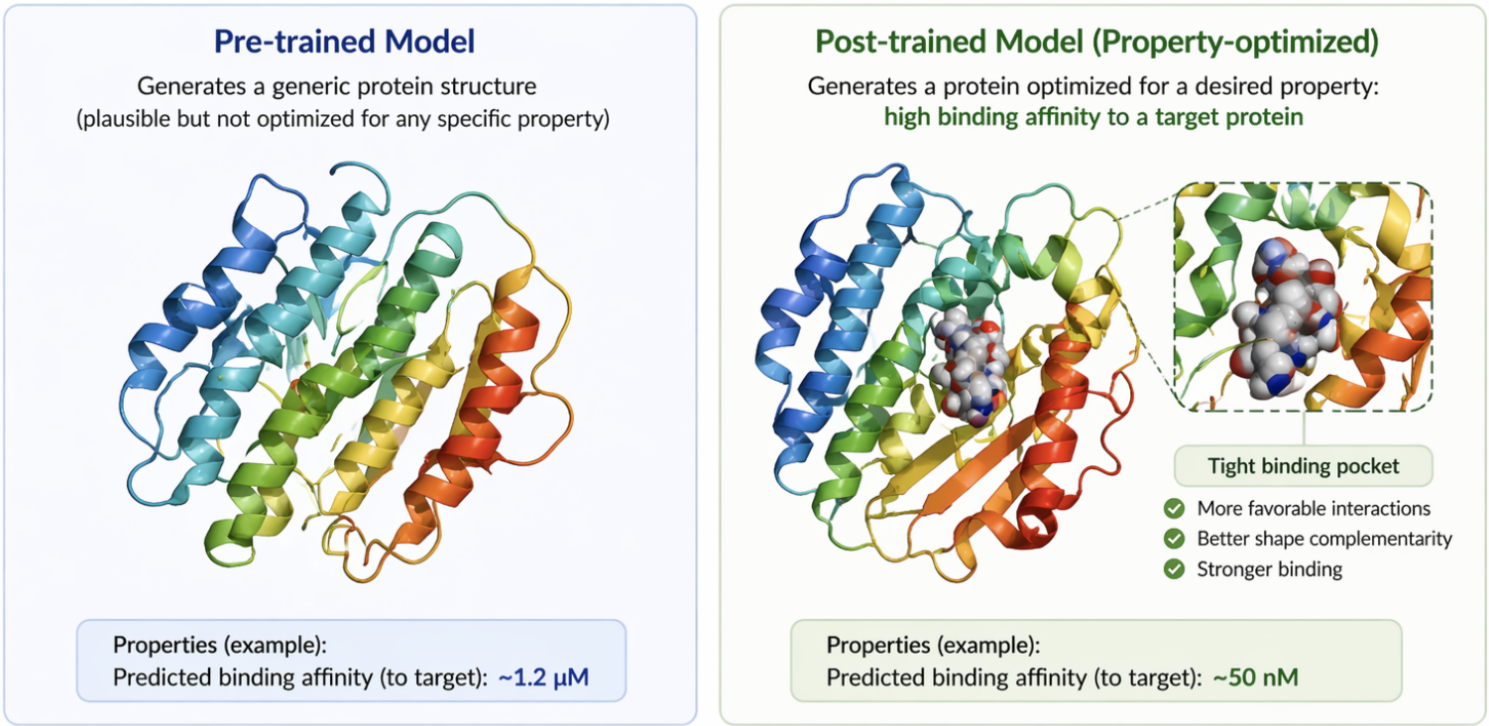
Generating a human-preferred image

- **Pre-trained model:** Generates an image of a flower in a bottle.
- **Post-trained model:** Generates a visually appealing image in a warm, inviting environment.



Generating protein structure with functional goals

- **Pre-trained model:** Generates protein structure that looks realistic
- **Post-trained model:** Generates a protein structure that satisfies properties such as high binding affinity



Why Reinforcement Learning for Diffusion Post-Training?

Optimize a downstream reward:

- easy-to-understand language
- preserves correctness while improving helpfulness

Why Reinforcement Learning for Diffusion Post-Training?

Optimize a downstream reward:

- easy-to-understand language
- preserves correctness while improving helpfulness

RL is a natural framework for post-training diffusion models

- Denoising process can be viewed as a sequential decision problem
- Reward can capture downstream objectives
- RL can optimize the expected rewards that are
  - **non-differentiable**,
  - **noisy**,
  - only available through **black-box evaluator**.

Why Reinforcement Learning for Diffusion Post-Training?

Optimize a downstream reward:

- easy-to-understand language
- preserves correctness while improving helpfulness

RL is a natural framework for post-training diffusion models

- Denoising process can be viewed as a sequential decision problem
- Reward can capture downstream objectives
- RL can optimize the expected rewards that are
  - **non-differentiable**,
  - **noisy**,
  - only available through **black-box evaluator**.
- Popular RL algorithms for post-training of diffusion models
  - DDPO (Black et al. 2023), DPOK (Fan et al. 2023), Diffusion-DPO (Wallace et al. 2023), DSPO (Zhang et al. 2024), ...

Questions We Want to Answer

- **Convergence:** Can PPO-style fine-tuning of diffusion models be proved to converge at all?

Questions We Want to Answer

- **Convergence:** Can PPO-style fine-tuning of diffusion models be proved to converge at all?
- **Sampler choice:** between DDPM-like stochastic samplers and DDIM-like deterministic samplers, which is more favorable for fine-tuning?

Questions We Want to Answer

- **Convergence:** Can PPO-style fine-tuning of diffusion models be proved to converge at all?
- **Sampler choice:** between DDPM-like stochastic samplers and DDIM-like deterministic samplers, which is more favorable for fine-tuning?
- **KL regularization:** how does the KL coefficient μ affect optimization stability and convergence speed?

These are the three main questions answered in this paper.

Forward Diffusion Process

- Continuous-time variance-preserving Ornstein–Uhlenbeck (OU) SDE:

$$dX_\tau = -\tfrac{1}{2} X_\tau \, d\tau + dW_\tau, \qquad X_0 = x_0.$$

- Density $p(x, \tau)$ satisfies the Fokker–Planck equation:

$$\partial_\tau p = \tfrac{1}{2} \nabla_x \cdot (xp) + \tfrac{1}{2} \Delta_x p.$$

Forward Diffusion Process

- Continuous-time variance-preserving Ornstein–Uhlenbeck (OU) SDE:

$$dX_\tau = -\tfrac{1}{2} X_\tau \, d\tau + dW_\tau, \qquad X_0 = x_0.$$

- Density $p(x, \tau)$ satisfies the Fokker–Planck equation:

$$\partial_\tau p = \tfrac{1}{2} \nabla_x \cdot (xp) + \tfrac{1}{2} \Delta_x p.$$

- Equivalent discrete-time formulation (Ho et al. 2020; Song et al. 2021):

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1 - \bar{\alpha}_t}\, z, \qquad z \sim \mathcal{N}(0, I),$$

with $\bar{\alpha}_t = \prod_{i=1}^{t} (1 - \beta_i)$. For time-homogeneous $\beta_t \equiv \beta$ and $\tau_t := \beta t$, $\bar{\alpha}_t = e^{-\tau_t}$, so $x_t \overset{d}{=} X_{\tau_t}$.

Reverse Process: A Unified Interpolated Sampler

- Reverse-time SDE driven by the score $\nabla_x \log p(x, \tau)$.

$$dY_\tau = -\left(\tfrac{1}{2} Y_\tau + \tfrac{1+\lambda}{2} \nabla_x \log p(Y_\tau, \tau)\right) d\tau + \sqrt{\lambda}\, d\widetilde{W}_\tau.$$

  - $\lambda = 1$: DDPM-like (fully stochastic SDE)
  - $\lambda = 0$: DDIM-like (PF-ODE).

Reverse Process: A Unified Interpolated Sampler

- Reverse-time SDE driven by the score $\nabla_x \log p(x, \tau)$.

$$dY_\tau = -\left(\tfrac{1}{2} Y_\tau + \tfrac{1+\lambda}{2} \nabla_x \log p(Y_\tau, \tau)\right) d\tau + \sqrt{\lambda}\, d\widetilde{W}_\tau.$$

  - $\lambda = 1$: DDPM-like (fully stochastic SDE)
  - $\lambda = 0$: DDIM-like (PF-ODE).

- Euler–Maruyama discretization with learned score s_θ:

$$\bar{x}_{t-1} = \bar{x}_t - \beta\left(\tfrac{1}{2}\bar{x}_t + \tfrac{1+\lambda}{2} s_\theta(\bar{x}_t, t)\right) + \sqrt{\lambda\beta}\, \xi_t,\ \xi_t \sim \mathcal{N}(0, I).$$

  - **Unified sampler:** interpolated trajectories between DDPM and DDIM.

Reverse Process: A Unified Interpolated Sampler

- Reverse-time SDE driven by the score $\nabla_x \log p(x, \tau)$.

$$dY_\tau = -\left(\tfrac{1}{2} Y_\tau + \tfrac{1+\lambda}{2} \nabla_x \log p(Y_\tau, \tau)\right) d\tau + \sqrt{\lambda}\, d\widetilde{W}_\tau.$$

  - $\lambda = 1$: DDPM-like (fully stochastic SDE)
  - $\lambda = 0$: DDIM-like (PF-ODE).

- Euler–Maruyama discretization with learned score s_θ:

$$\bar{x}_{t-1} = \bar{x}_t - \beta\left(\tfrac{1}{2}\bar{x}_t + \tfrac{1+\lambda}{2} s_\theta(\bar{x}_t, t)\right) + \sqrt{\lambda\beta}\, \xi_t,\ \xi_t \sim \mathcal{N}(0, I).$$

  - **Unified sampler:** interpolated trajectories between DDPM and DDIM.

- **Pretraining conditions (standard, mild):**
  - Finite L^2 log-density moment and second moment of X_0.
  - Pretrained score accuracy along reverse trajectories:

$$\mathbb{E}_{\omega \sim P_{\theta_{\text{pre}}}} \sum_{t=1}^{T} \left\| s_{\theta_{\text{pre}}}(\bar{x}_t, t) - \nabla \log p_t(\bar{x}_t, \tau_t) \right\|^2 \le \bar{\varepsilon}^2.$$

RL Fine-Tuning Objective

- View reverse sampling as a finite-horizon MDP: state $\bar{x}_t$, action = score-induced drift.
- A trajectory $\omega := (\bar{x}_T, \ldots, \bar{x}_0) \sim P_\theta$, with terminal reward $r(\omega)$, $|r(\omega)| \leq R$.

RL Fine-Tuning Objective

- View reverse sampling as a finite-horizon MDP: state $\bar{x}_t$, action = score-induced drift.
- A trajectory $\omega := (\bar{x}_T, \ldots, \bar{x}_0) \sim P_\theta$, with terminal reward $r(\omega)$, $|r(\omega)| \leq R$.
- **KL-regularized trajectory-level objective:**

$$J(\theta) := \mathbb{E}_{\omega \sim P_\theta}[r(\omega)] - \mu\, \mathcal{D}(\theta), \qquad \mathcal{D}(\theta) := \mathrm{KL}\big(P_\theta \,\|\, P_{\theta_{\mathrm{pre}}}\big)\,.$$

Trajectory-Level KL Reduces to a Score L_2 Gap

- With Euler–Maruyama, the Markov transition is Gaussian:

$$p_\theta(\bar{x}_{t-1} \mid \bar{x}_t) = \mathcal{N}\big(\mu_\theta(\bar{x}_t, t),\ \lambda\beta\, I\big),$$

where $\mu_\theta(\bar{x}_t, t) = \bar{x}_t - \beta\big(\frac{1}{2}\bar{x}_t + \frac{1+\lambda}{2} s_\theta(\bar{x}_t, t)\big)$.

Trajectory-Level KL Reduces to a Score L_2 Gap

- With Euler–Maruyama, the Markov transition is Gaussian:

$$p_\theta(\bar{x}_{t-1} \mid \bar{x}_t) = \mathcal{N}\big(\mu_\theta(\bar{x}_t, t),\ \lambda\beta\, I\big),$$

  where $\mu_\theta(\bar{x}_t, t) = \bar{x}_t - \beta\big(\frac{1}{2}\bar{x}_t + \frac{1+\lambda}{2} s_\theta(\bar{x}_t, t)\big)$.
- Same $\lambda\beta I$ covariance $\Rightarrow$ trajectory-level KL becomes a weighted squared-ℓ_2 gap:

$$\mathcal{D}(\theta) = \frac{\beta(1+\lambda)^2}{8\lambda}\, \mathbb{E}_{\omega\sim P_\theta}\left[\sum_{t=1}^{T} \left\| s_\theta(\bar{x}_t, t) - s_{\theta_{\text{pre}}}(\bar{x}_t, t)\right\|^2\right].$$

  - This **KL-to-L_2 link** makes the role of λ and μ explicit in the analysis.

PPO as Batched Online Policy Optimization

- PPO is an on-policy but batched policy optimization method:
  - Using the current policy θ_{old}, collect a batch of trajectories
  - Reuse this same batch for several local gradient updates.
- **Why use batches?**
  - Trajectory sampling is expensive.
  - Reusing one batch for multiple updates improves sample efficiency.
- **Why only several updates?**
  - After too many updates, the current policy θ drifts too far from the behavior policy θ_{old},
  - The batch no longer accurately represents the new policy.
- **PPO philosophy:**

$$\text{collect trajectories} \rightarrow \text{do several local updates} \rightarrow \text{resample.}$$

Importance Ratio and Clipping

- A batch of trajectories are distributed as $\omega \sim P_{\theta_{\text{old}}}$
- The policy has been improved to θ, whose trajectory law is P_θ.
- PPO surrogate loss with KL penalty:

$$L(\theta) := \mathbb{E}_{\omega \sim P_{\theta_{\text{old}}}} \left[b(\omega; \theta, \theta_{\text{old}}) \big(r(\omega) - \mu \, \widehat{\mathcal{D}}(\omega; \theta) \big) \right].$$

- This introduces trajectory likelihood ratio:

$$b(\omega; \theta, \theta_{\text{old}}) := dP_\theta / dP_{\theta_{\text{old}}}(\omega)$$

- PPO clips ratio to mitigate high variance in policy-gradient estimator:

$$\bar{b}(\omega; \theta, \theta_{\text{old}}) := \text{Clip}(b(\omega; \theta, \theta_{\text{old}}), \delta),$$

where $\text{Clip}(x, \delta) \in \{1 - \delta, \, x, \, 1 + \delta\}$ as in standard PPO.

PPO Fine-Tuning of Diffusion Samplers

- **Inputs:** initial θ_0, reference θ_{pre}, batch G, mini-batch B, stepsize η, outer iters K, inner updates U.
- **For** $k = 0, \ldots, K-1$:
  - Set $\theta_{\mathrm{old}} \leftarrow \theta_k$; sample G trajectories $\{\omega_i\}_{i=1}^{G} \sim P_{\theta_{\mathrm{old}}}$; compute rewards.
  - Initialize $\theta_{k,0} \leftarrow \theta_k$.
  - **For** $u = 0, \ldots, U-1$:
    - Sample indices $\{i_b\}_{b=1}^{B}$ i.i.d. from $\{1, \ldots, G\}$.
    - Compute trajectory-level KL proxy $\widehat{\mathcal{D}}(\omega_{i_b}; \theta_{k,u})$.
    - Form mini-batch surrogate loss; update $\theta_{k,u+1} \leftarrow \theta_{k,u} + \eta \nabla \widehat{L}(\theta_{k,u})$.
  - Set $\theta_{k+1} \leftarrow \theta_{k,U}$.
- For analysis, flatten (k, u) to a global index $m := kU + u$.

Technical Assumptions

Assumption 1 (Local regularity & Fisher nondegeneracy)

s_θ is C^2; $\exists\, \ell, C_\ell, F_\ell > 0$ s.t. for all θ with $\|\theta - \theta_{\mathrm{pre}}\| \le \ell$:

(i) $\|\nabla_\theta s_\theta(\bar{x}_t, t)\| \le C_\ell(1 + \|\bar{x}_t\|)$ *for all* $(\bar{x}_t, t)$;

(ii) *Trajectory-level Fisher information matrix*

$$F(\theta) := \mathbb{E}_{\omega\sim P_\theta}\big[\nabla_\theta \log P_\theta(\omega)\, \nabla_\theta \log P_\theta(\omega)^\top\big] \succeq F_\ell\, I_d$$

- (i) sensitivity of score to parameter grows at most linearly with $\|\bar{x}_t\|$
- (ii) KL geometry controls Euclidean parameter shift
- **This is mild:** local on the ℓ-ball only; outside, no requirement on s_θ

Assumption 2 (Global regularity of the reverse drift)

Let $b_\theta(x,\tau) = -\frac{1}{2}x - \frac{1+\lambda}{2}\nabla_x \log p(x,\tau)$. $\exists\, L > 0$, $\forall \theta, x, y, \tau, \tau'$:

$$\|b_\theta(x,\tau) - b_\theta(y,\tau)\| \le L\|x - y\|, \quad \|b_\theta(x,\tau) - b_\theta(x,\tau')\| \le L\,|\tau - \tau'|^{1/2}.$$

Main Result: Convergence of PPO Fine-Tuning

Theorem 1 (Convergence of PPO for diffusion fine-tuning)

Let $\beta = c/T$, $\mu \geq \frac{16}{F_l l^2}(R - J(\theta_{\text{pre}}))$, $\rho := 1 - \frac{\beta}{2} + \frac{(1+\lambda)L_{xx}\beta}{2}$,

$$\eta = \min\left\{ \frac{1}{\sqrt{N}},\ \frac{\mu F_l l^2 h \lambda^{3/2}}{\sqrt{2C_1 N}(1+\lambda)^2\sqrt{1+\rho^{2T}}},\ \frac{\sqrt{h}\, l\, \lambda^{3/2}}{\sqrt{C_3 N}(1+\lambda)^2\sqrt{1+\rho^{2T}}} \right\},$$

Then with probability at least $1 - 2h$*:*

$$\frac{1}{N}\sum_{m=1}^{N-1} \|\nabla J(\theta_m)\|^2 \leq \mathcal{O}(1)\,\frac{(1+\lambda)^2\sqrt{1+\rho^{2T}}}{h\mu\lambda^{3/2}\sqrt{N}} + \mathcal{O}(1)\,\frac{(1+\lambda)^4(1+\rho^{2T})}{h\lambda^3\sqrt{N}}.$$

- **First term:** variance of the policy-gradient estimator.
- **Second term:** clipping error
- Both decay in $O(\frac{1}{\sqrt{N}}) \Rightarrow$ convergence to a stationary point.

More Stochasticity Helps

The Suboptimality Gap depends on

$$f_1(\lambda) := \frac{(1+\lambda)^2\sqrt{1+\rho^{2T}}}{\lambda^{3/2}}, \qquad f_2(\lambda) := \frac{(1+\lambda)^4(1+\rho^{2T})}{\lambda^3},$$

More Stochasticity Helps

The Suboptimality Gap depends on

$$f_1(\lambda) := \frac{(1+\lambda)^2\sqrt{1+\rho^{2T}}}{\lambda^{3/2}}, \qquad f_2(\lambda) := \frac{(1+\lambda)^4(1+\rho^{2T})}{\lambda^3},$$

Both functions monotonically decreasing in $\lambda \in (0, 1]$. $\lambda = 1$ results in fastest convergence.

More Stochasticity Helps

The Suboptimality Gap depends on

$$f_1(\lambda) := \frac{(1+\lambda)^2\sqrt{1+\rho^{2T}}}{\lambda^{3/2}}, \qquad f_2(\lambda) := \frac{(1+\lambda)^4(1+\rho^{2T})}{\lambda^3},$$

Both functions monotonically decreasing in $\lambda \in (0, 1]$. $\lambda = 1$ results in fastest convergence.

Increased sampler stochasticity λ, which corresponds to trajectories closer to DDPM-style sampling, is more favorable for RL fine-tuning.

Intuitive Reason: Greater stochasticity enhances exploration in RL, enabling faster reward maximization.

Beyond Interpolation Regime

$$\bar{x}_{t-1} = \bar{x}_t - \beta\left(\frac{1}{2}\bar{x}_t + \frac{1+\lambda}{2}s_\theta(\bar{x}_t, t)\right) + \sqrt{\lambda\beta}\,\xi_t,\ \xi_t \sim \mathcal{N}(0, I).$$

- $\lambda > 1$:
  - More stochastic than the standard reverse-SDE/DDPM case
  - Each reverse step adds more noise than the usual reverse SDE sampler
  - Stronger score-driven denoising to compensate for the extra noise

Beyond Interpolation Regime

$$\bar{x}_{t-1} = \bar{x}_t - \beta\left(\tfrac{1}{2}\bar{x}_t + \tfrac{1+\lambda}{2} s_\theta(\bar{x}_t, t)\right) + \sqrt{\lambda\beta}\,\xi_t,\ \xi_t \sim \mathcal{N}(0, I).$$

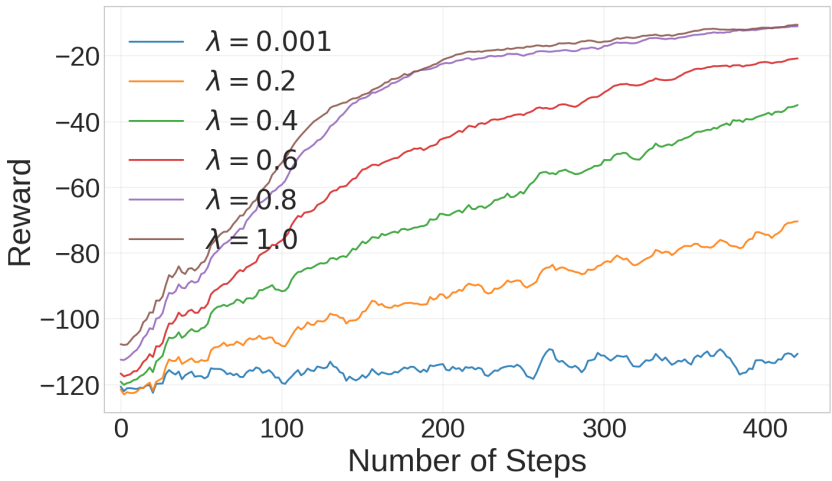
- $\lambda > 1$:
  - More stochastic than the standard reverse-SDE/DDPM case
  - Each reverse step adds more noise than the usual reverse SDE sampler
  - Stronger score-driven denoising to compensate for the extra noise

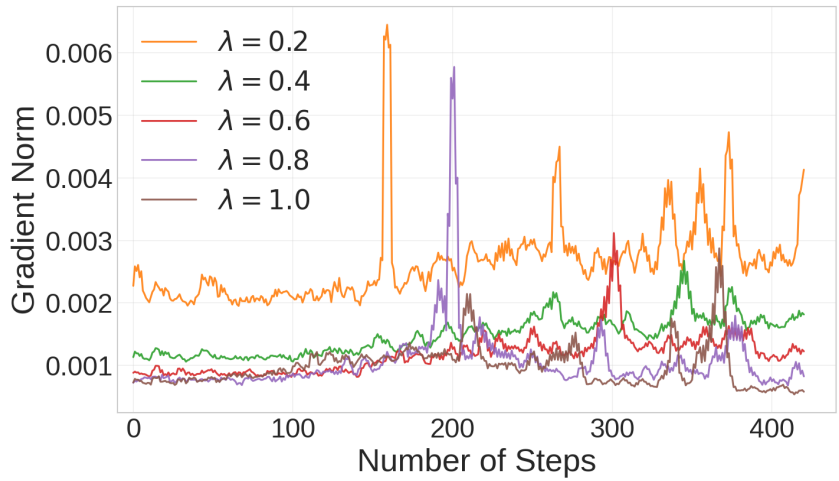
There exists an optimal $\lambda^* \in [1, 3]$ that minimizes the suboptimality gap, resulting in the fastest convergence.

Overly broad exploration injects excessive noise into the policy gradient and ultimately hinders fine-tuning convergence.

Experiments: Effect of Sampler Stochasticity λ

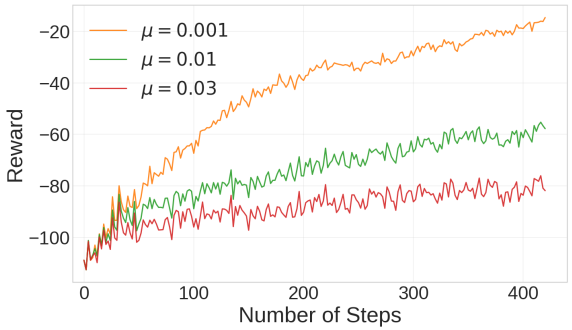
- Task: compressibility — reward encourages images with smaller JPEG file sizes.
- Fine-tuning pipeline following Black et al. (2023); base model: Stable Diffusion v1.5.

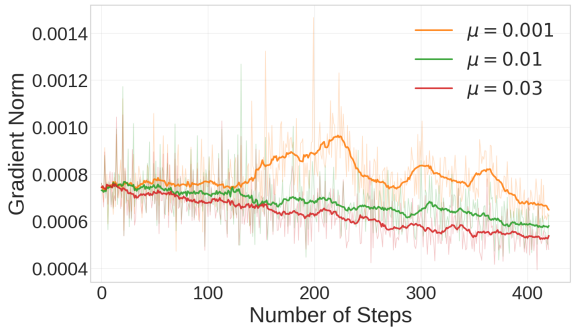


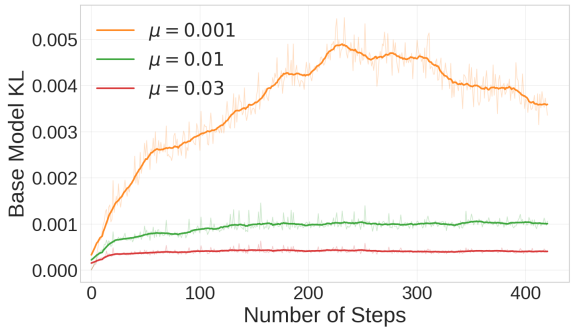


- **Reward (left):** larger $\lambda \Rightarrow$ faster reward improvement
- **Gradient norm (right):** larger $\lambda \Rightarrow$ smaller gradient magnitude

Experiments: Effect of KL Coefficient μ







- **Reward (left):** small μ allows larger deviation and more reward gain.
- **Gradient norm (middle):** larger μ pushes training into a more stable regime earlier; gradients decay faster and stay smaller.
- **KL to base (right):** larger μ suppresses drift from θ_{base}; the KL curve flattens earlier.
- **Trade-off (main theorem):** larger μ yields more conservative generation and slower reward improvement, but better stability.

Theoretical Challenges

Challenge 1: Controlling policy-gradient variance for trajectories.

- PPO gradients depend on trajectory likelihoods, whose variance is governed by correlated denoising steps along the reverse path.
- **Technique:** Exploits *martingale-difference decomposition* of trajectory score to cancel cross-time correlations and control gradient variance.

Proposition 1 (Martingale difference)

Factor $\nabla \log \varphi_\theta(\omega) = \sum_{t=1}^{T} \Delta_t$, *where* $\Delta_t := \nabla \log \varphi_\theta(\bar{x}_{t-1} \mid \bar{x}_t)$. *Let* $\mathcal{G}_t := \sigma(\bar{x}_T, \ldots, \bar{x}_t)$. *Then*

$$\mathbb{E}_{\omega \sim P_{\theta_{k,u}}}\big[\Delta_t \mid \mathcal{G}_t\big] = 0, \qquad t = 1, \ldots, T.$$

Theoretical Challenges

Challenge 2: Trajectory-level distribution shift under denoising.

- A policy update changes law of entire stochastic trajectory
- **Technique:** Derives trajectory-level L^2 stability

Proposition 2 (One-step update bound)

When $\|\theta_m - \theta_{\text{pre}}\| \le \ell$,

$$\mathbb{E}[\|\theta_{m+1} - \theta_m\|^2 \mid \mathscr{F}_m] \le \mathcal{O}(1)\eta^2 \tfrac{(1+\lambda)^2}{\lambda} \beta \, \mathbb{E} \sum_{t=1}^{T} (1 + \|\bar{x}_t\|^2).$$

Proposition 3 (Uniform trajectory second-moment bound)

Suppose $\|\theta - \theta_{\text{pre}}\| \le \ell$ *and let* $\beta = c/T$ *and* $\rho := 1 - \frac{\beta}{2} + \frac{(1+\lambda)L_{xx}\beta}{2}$. *Then*

$$\beta \sum_{t=0}^{T} \mathbb{E}\|\bar{x}_t\|^2 \le \mathcal{O}(1)(1+\lambda)^2(1+\rho^{2T})\big(1 + \bar{\varepsilon}^2 + \mathcal{D}(\theta)\big).$$

RHS is bounded as T becomes large, stablizing post-training even for large number of diffusion steps.

Technical Challenges

Challenge 3: Handling PPO-specific clipping and keeping iterates in a valid local region.

- Assumptions only hold locally around the pretrained model
- **Technique:** *KL-regularized localization + stopping-time argument*

Proof Sketch: Stopping Time & Localization

Define

$$\tau := \inf\{m \geq 1 : \|\theta_m - \theta_{\text{pre}}\| > \ell/2\}, \qquad \tau_N := \tau \wedge N,$$

and

$$\mathcal{E}_{\text{last},N} := \{\tau > N\} \cup \big(\{\tau \leq N\} \cap \{\|\theta_\tau - \theta_{\text{pre}}\| \leq \ell\}\big).$$

Every stopped update up to time $\tau_N - 1$ has both endpoints in the local ℓ-ball, so all local smoothness and Fisher-type controls remain valid along the stopped path.

Step 1: Descent Lemma & Error Decomposition

Let $\|\theta_m - \theta_{\text{pre}}\| \le \ell$ and $\|\theta_{m+1} - \theta_{\text{pre}}\| \le \ell$. The following holds:

$$\begin{aligned}&(R - \mathcal{J}(\theta_{m+1})) - (R - \mathcal{J}(\theta_m)) \\ &\quad \le -\eta \|\nabla \mathcal{J}(\theta_m)\|^2 + \mathrm{Err}_m + D_m + \eta^2 (L_{J,\ell} + L_{D,\ell}) \|\theta_{m+1} - \theta_m\|^2\end{aligned}$$

- **Clipping remainder**: Err_m is the squared-norm remainder coming from trajectories whose likelihood ratio exits clipping window.
- **Gradient-noise term**: $D_m := \eta \nabla \mathcal{J}(\theta_m)^\top (\nabla \mathcal{J}(\theta_m) - \widehat{\nabla} \mathcal{J}(\theta_m))$
  - $\widehat{\nabla} \mathcal{J}(\theta_m)$: unbiased main estimator built from unclipped leading terms.
- **Assumption 1** implies that inside the closed ℓ-ballmodel is locally C^2, so J and D are twice continuously differentiable there; on this compact set, Hessian norms attain finite maxima, which define $L_{J,\ell}, L_{D,\ell}$.

Step 2: Bounding Cumulative Errors

Clipping Error: Bound the event of large importance ratio via $\|\theta_m - \theta_{k_m}\|$, and then apply Propositions 2 and 3.

$$H_1 := \mathbb{E}\left[\sum_{m=1}^{\tau_N - 1} \mathbb{E}[\mathrm{Err}_m \mid \mathscr{F}_m]\right] \leq C_1 N\eta^2 \frac{(1+\lambda)^4}{\delta_0 \lambda^3}(1+\rho^{2T}).$$

One-Step Update Error: by Propositions 2 and 3

$$\begin{aligned} H_2 &:= \eta^2(L_{J,I} + L_{D,I})\mathbb{E}\left[\sum_{m=1}^{\tau_N - 1} \mathbb{E}[\|\theta_{m+1} - \theta_m\|^2 \mid \mathscr{F}_m]\right], \\ &\lesssim N\eta^2 \frac{(1+\lambda)^4}{\lambda^3}(1+\rho^{2T}) \end{aligned}$$

Gradient Noise: D_m is a martingale difference sequence and bound gradient moments by $\|\theta_m - \theta_{k_m}\|$, and then apply Propositions 2 and 3.

$$\begin{aligned} H_3 &:= \mathbb{E}^{1/2}\left[\sum_{m=1}^{\tau_N - 1} \mathbb{E}[D_m^2 \mid \mathscr{F}_m]\right] \\ &\leq \frac{F_I I^2 \mu}{16} \cdot \frac{h}{2} + \mathcal{O}(1)\frac{1}{\mu h}\eta^2 \frac{(1+\lambda)^4}{\lambda^3}(1+\rho^{2T}) \end{aligned}$$

Step 3: Probability of Escape Neighborhood

- Let $I_N := 1_{\mathcal{E}_{\text{last},N}} 1_{[\tau \le N]}$, escaping with small enough exit step
- Due to escaping fact $\|\theta_\tau - \theta_{\text{pre}}\| > l/2$, Assumption 1(ii) of KL-to-Euclidean control, and $\mu \ge \frac{16}{F_l l^2}(R - J(\theta_{\text{pre}}))$

$$I_N\big(R - J(\theta_{\tau_N}) - (R - J(\theta_{\text{pre}}))\big) \ge \frac{F_l l^2 \mu}{16} I_N$$

- Telescoping descent lemma, and taking expectation

$$\frac{F_l l^2 \mu}{16} \mathbb{P}(\mathcal{E}_{\text{last},N} \cap (\tau \le N)) \le H_1 + H_2 + H_3.$$

- On $\{\tau \le N\} \cap \mathcal{E}^c_{\text{last},N}$, $\|\theta_{\tau_N} - \theta_{\tau_N - 1}\| > l/2$, which yields

$$\mathbb{P}\big(\{\tau \le N\} \cap \mathcal{E}^c_{\text{last},N}\big) \le \frac{4C_3}{l^2}\eta^2 \frac{(1+\lambda)^4}{\lambda^3}(1 + \rho^{2T}).$$

- Combining the two bounds with a good stepsize gives

$$\mathbb{P}(\tau \le N) \le h.$$

Step 4: Convergence in Gradient Norm

- Define $J_N := 1_{\mathcal{E}_{\text{last},N}}$ and telescope descent lemma

$$\frac{\eta}{4}\mathbb{E}\left[J_N \sum_{m=1}^{\tau_N-1} \|\nabla J(\theta_m)\|^2\right] \leq R - J(\theta_{\text{pre}}) + H_1 + H_2 + H_3.$$

- The above bound and Markov's inequality imply with probability at least $1-h$,

$$\frac{1}{N} J_N \sum_{m=1}^{\tau_N-1} \|\nabla J(\theta_m)\|^2 \leq \mathcal{O}(1)\,\frac{(1+\lambda)^2\sqrt{1+\rho^{2T}}}{h\mu\lambda^{3/2}\sqrt{N}} + \mathcal{O}(1)\,\frac{(1+\lambda)^4(1+\rho^2$$

- On $\{\tau > N\}$ (which has probability greater than $1-h$), $J_N = 1$ and $\tau_N = N$ in the above equation
- Convergence follows w.p. $\geq 1-2h$.

Conclusion

- We establish the first convergence guarantee for PPO-style RL fine-tuning of diffusion models.
- **Insights:**
  - Sampler stochasticity $\lambda \in (0, 1]$: closer to $\lambda = 1$ (DDPM) $\Rightarrow$ faster convergence. The DDIM endpoint $\lambda 0$ slow convergence; Increasingly large λ beyond 1 can hurt convergence
  - KL coefficient μ: it governs trajectory-level drift from the pretrained sampler; larger μ yields more conservative and earlier-stabilizing updates, with a reward-speed trade-off.

Conclusion

- We establish the first convergence guarantee for PPO-style RL fine-tuning of diffusion models.
- **Insights:**
  - Sampler stochasticity $\lambda \in (0, 1]$: closer to $\lambda = 1$ (DDPM) $\Rightarrow$ faster convergence. The DDIM endpoint $\lambda 0$ slow convergence; Increasingly large λ beyond 1 can hurt convergence
  - KL coefficient μ: it governs trajectory-level drift from the pretrained sampler; larger μ yields more conservative and earlier-stabilizing updates, with a reward-speed trade-off.
- **Future Work:**
  - Impact of the choise of T
  - Identify favorable parameterization for policy that yeilds global convergence
  - Extension to discrete diffusion models