Amortized Guidance for Image Inpainting with Pretrained Diffusion Models

Yilie Huang
Department of
Applied Mathematics
The Hong Kong
Polytechnic University
Hong Kong
yilie.huang@polyu.edu.hk

Xun Yu Zhou
Department of Industrial
Engineering and
Operations Research and
Data Science Institute
Columbia University
New York, NY 10027, USA
xz2574@columbia.edu

Abstract

We study image inpainting with generative diffusion models. Existing methods typically either train dedicated task-specific models, or adapt a pretrained diffusion model separately for each masked image at deployment. We introduce a middle-ground model, termed Amortized Inpainting with Diffusion (AID), which keeps a pretrained diffusion backbone fixed, trains a small reusable guidance module offline, and then reuses it across masked images without per-instance optimization. We formulate it as a deterministic guidance problem with a supervised terminal objective. To make this problem learnable in high dimensions, we derive an auxiliary Gaussian formulation and prove that solving this randomized problem recovers the optimal deterministic guidance field. This bridge yields a principled continuous-time actor–critic algorithm for learning the guidance module in a fully data-driven manner. Empirically, on AFHQv2 and FFHQ under the pixel EDM pipeline and on ImageNet under the latent EDM2 pipeline, AID consistently improves the quality–speed trade-off over strong fixed-backbone and amortized inpainting baselines across multiple mask types, while adding less than one percent trainable overhead.

1 Introduction

Image inpainting is a conditional generation problem: given a partially observed image (e.g. a damaged photo), the goal is to complete the missing region in a way that is both visually plausible and consistent with the visible pixels. Recent diffusion models [10, 37, 33, 34] have made this task substantially more effective, especially when strong pretrained generative models are available. This makes inpainting a particularly important setting for studying how a powerful pretrained generative model can be adapted to conditional tasks efficiently and accurately.

Despite this progress, the current research remains unsatisfactory. Existing approaches are largely split between two directions. One is to train a dedicated inpainting model from scratch or a substantial task-specific adaptation module, as in methods such as LaMa [40], MAT [29], Palette [36], BrushNet [21], and DAVI [27]. This direction amortizes inference, but does so by adding large learned components and requiring substantial training time. The other route is to keep a pretrained diffusion model fixed and adapt it separately for each masked image at test time, as in methods such as RePaint [31], DPS [3], and MCG [2]. This preserves the pretrained model, but each masked image is still treated as a new inference problem, leading to repeated case-by-case optimization or iterative refinement during deployment. In other words, one direction amortizes well but pays in learned-model complexity,

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

while the other reuses the pretrained model but pays repeatedly at test time. There is a gap in the middle ground.

Our answer to this gap is an *Amortized Inpainting with Diffusion* (AID) model, which keeps the pretrained diffusion model fixed and trains only a small reusable guidance module offline. Once trained, the same module is reused across masked images at test time; so inference no longer requires solving a new optimization problem for every task. AID is built on a controlled-guidance view of inpainting: the guidance deployed for actual tasks is deterministic, while learning uses randomized Gaussian policies. A central contribution of the paper is to prove that this randomized learning problem is optimizer-preserving, making the resulting actor–critic algorithm theoretically grounded rather than heuristic.

The main contributions of this paper are as follows:

- *Methodology*: We propose AID, a controlled-guidance framework for image inpainting that keeps a pretrained diffusion model fixed and trains offline a lightweight guidance module once. The learned module is then reused across masked images at test time, combining the reuse advantage of pretrained diffusion models with the efficiency of amortized inference.
- *Theory*: We develop a rigorous theoretical foundation for AID by establishing a policy-equivalence bridge between the deterministic guidance problem and an auxiliary Gaussian actor–critic formulation. Specifically, we prove both directions of the correspondence: the deterministic optimal guidance is sufficient to construct an optimal auxiliary Gaussian policy, and any optimal auxiliary Gaussian policy must recover this deterministic guidance through its mean. This optimizer-preserving bridge justifies learning with randomized Gaussian policies while deploying the deterministic policy mean.
- *Experiments*: We evaluate AID on AFHQv2, FFHQ, and ImageNet under multiple mask types and metrics, with well under 1% trainable overhead relative to the pretrained diffusion model. Across fixed-backbone baselines, the default setting consistently delivers the strongest performance and surpasses RePaint using only about 10% of its sampling budget in both pixel and latent spaces. The same trained guidance module transfers without retraining to unseen masks and to a lower-latency sampler, which further reduces sampling time while still outperforming all non-RePaint baselines and remaining competitive with RePaint.

To our best knowledge, AID is the first amortized controlled-guidance framework for image inpainting that keeps the pretrained diffusion backbone frozen, trains only a small reusable guidance module, and provides an exact optimizer-preserving bridge from stochastic actor–critic learning to the deterministic guidance field used at deployment. For complete proofs, implementation details, additional quantitative and qualitative results, and ablations, see the extended version [13].

Relevant literature: In addition to the existing inpainting research discussed earlier on two major directions, we mention a related control-based work that largely follows the second direction. [28] formulates inpainting as a test-time control problem, but it still requires instance-wise iterative optimization and incurs substantial sampling cost. The closest middle-ground comparator is LatentPaint [4], a practical propagation-based method for reusable inpainting adaptation. While it shares our goal of amortized adaptation, it is tied to latent-space updates. AID instead learns principled reusable guidance and applies to both pixel-space EDM and latent-space EDM2 backbones.

Continuous-time reinforcement learning (CT-RL) provides the methodological foundation for our learning formulation, while the optimizer-preserving Gaussian bridge developed below is specific to AID. Since modern diffusion models for generative AI are naturally described through continuous-time SDEs and ODEs [38], CT-RL is a particularly natural framework for our problem. This line of work was initiated by Wang et al. [43] and has since developed into a model-free theory based on martingale methods [19, 20, 41], with further results on performance guarantees and data-driven exploration [15, 12]. The framework has also been applied to portfolio selection and asset–liability management [14, 16, 11]. More recently, CT-RL has also begun to appear in diffusion model training, fine-tuning, and timestep selection [8, 46, 17, 18]. Recent work on diffusion fine-tuning also adapts pretrained diffusion models toward desired objectives or conditioning information [42, 6]. AID instead focuses on supervised image inpainting and learns a small reusable guidance module while keeping the diffusion backbone fixed. A related discrete-time RL-based inpainting work is PrefPaint [30], which fine-tunes the model using large-scale human preference annotations and thus relies on human-driven, subjective supervision rather than our purely data-driven setting.

Organization: Section 2 formulates amortized inpainting as a controlled diffusion problem. Section 3 presents the optimizer-preserving Gaussian bridge, and Section 4 turns it into the AID actor–critic solver. Section 5 reports experiments.

2 Control Formulation for Amortized Inpainting

Setup. We first introduce the problem of amortized inpainting over a task distribution. Let $x^\dagger \in \mathbb{R}^d$ denote a clean reference image (from a training set), $M \in \mathbb{R}^{d\times d}$ be a diagonal binary mask matrix selecting the visible region, and $I - M$ select the missing region. We assume that $(x^\dagger, M)$ is sampled from a task distribution ρ_{task}, and define the masked observation by $y := Mx^\dagger$. The observable input is then $\xi := (M, y)$, while the full supervised tuple is $\zeta := (\xi, x^\dagger) = (M, y, x^\dagger)$. Thus, ξ is the information available for a specific inpainting task, whereas ζ is available only during offline training. Throughout, $\|\cdot\|$ denotes the Euclidean norm on $\mathbb{R}^d$. The key information structure is that deployment uses only ξ, while offline training may use ζ.

Let $t \in [0, T]$ denote diffusion time, where the forward process follows

$$\mathrm{d}\bar{x}(t) = -f(t)\bar{x}(t)\,\mathrm{d}t + g(t)\,\mathrm{d}w(t), \qquad \bar{x}(0) \sim p_0,$$

where p_0 is the data distribution of clean images. Let p_t denote the law of $\bar{x}(t)$ and let $S(t, x) = \nabla_x \log p_t(x)$ be the score. In our setting, S is replaced by a pretrained and fixed approximation $\hat{S}$. Then the associated probability-flow ordinary differential equation [38] is

$$\dot{x}(t) = b(t, x), \qquad x(0) \sim p_T,$$

where the reverse drift is defined by $b(t, x) := f(T - t)x + \frac{1}{2}g^2(T - t)\hat{S}(T - t, x)$.

Guided reverse dynamics. The drift $b(t, x)$ captures the frozen reverse dynamics induced by the pretrained diffusion model. For inpainting, however, this drift alone does not explicitly enforce consistency with the masked observation. Motivated by the general idea of diffusion guidance [7, 9, 32], we add a task-dependent control term for guidance and consider

$$\dot{x}(t) = b(t, x(t)) + u(t, x(t); \xi), \qquad x(t_0) = x_0, \tag{1}$$

where u is the guidance field. The information structure in (1) is that the deployed guidance law may depend on the mask and visible pixels, $\xi = (M, y)$, but not on the hidden clean image $x^\dagger$.

Terminal objective. For a supervised tuple $\zeta = (M, y, x^\dagger)$, we define the terminal loss by

$$\Psi(x; \zeta) = \alpha_{\text{vis}}\psi_{\text{vis}}(x; \zeta) + \alpha_{\text{hole}}\psi_{\text{hole}}(x; \zeta),$$

where $\alpha_{\text{vis}}, \alpha_{\text{hole}} \geq 0$ balance visible-region fidelity and missing-region fidelity,

$$\psi_{\text{vis}}(x; \zeta) := \frac{1}{2}\left\|M(x - x^\dagger)\right\|^2, \qquad \psi_{\text{hole}}(x; \zeta) := \frac{1}{2}\left\|(I - M)(x - x^\dagger)\right\|^2.$$

The first term enforces agreement with the observed pixels, while the second supervises the missing region using the clean image $x^\dagger$. Since the deployed guidance law depends only on the observable input $\xi = (M, y)$, the corresponding terminal target in the control problem is the conditional expectation $\bar{\Psi}(x; \xi) := \mathbb{E}[\Psi(x; \zeta) \mid \xi]$. During offline training, sampled supervised tuples ζ provide Monte Carlo realizations of this conditional target.

Control objective. Let $\mathcal{U}(\xi)$ denote the class of admissible feedback control fields $(t, x) \mapsto u(t, x; \xi) \in \mathbb{R}^d$. For $u \in \mathcal{U}(\xi)$, define

$$J^u(t, x; \xi) := \bar{\Psi}\big(x^u(T; t, x, \xi); \xi\big) + \int_t^T \frac{\beta}{2}\,\|u(r, x^u(r; t, x, \xi); \xi)\|^2\,\mathrm{d}r, \tag{2}$$

where $\beta > 0$ is a scalar control weight. The terminal term evaluates final inpainting quality averaged over the supervision with ξ, while the running term penalizes the cumulative strength of the guidance field along the trajectory. The latter reflects the notion that the pretrained reverse dynamics already provide a strong image model, and the additional guidance should intervene only when necessary to enforce consistency with the inpainting task. Although training is over samples from the task

distribution, the deployed controller observes the realized masked input ξ to deploy a control. Accordingly, the underlying control problem is formulated conditionally on ξ:

$$V(t,x;\xi) := \inf_{u\in\mathcal{U}(\xi)} J^u(t,x;\xi). \tag{3}$$

Under standard regularity assumptions ensuring well-posedness, V is the unique classical solution of

$$V_t(t,x;\xi) + \inf_{u\in\mathbb{R}^d}\left\{\nabla_x V(t,x;\xi)^\top\big(b(t,x)+u\big) + \frac{\beta}{2}\|u\|^2\right\} = 0, \tag{4}$$

with terminal condition

$$V(T,x;\xi) = \bar{\Psi}(x;\xi). \tag{5}$$

This completes the control formulation of our Amortized Inpainting with Diffusion (AID) model.

3 A Theoretical Bridge from Deterministic Guidance to Continuous-Time RL

Section 2 formulates AID as a deterministic control problem whose optimizer is the guidance used at deployment. The central challenge is that this problem is inherently high dimensional and induced by a highly nonlinear pretrained diffusion model; so the solution is generally not tractable analytically nor feasible numerically. To make this control problem tractable, we introduce an auxiliary *randomized* control problem with a carefully chosen Gaussian policy class.

This auxiliary formulation is useful only if it preserves the original deterministic control target. In particular, randomization should not change the guidance field ultimately deployed for inpainting. This section establishes this relationship: we show that solving the auxiliary randomized problem recovers exactly the deterministic optimal guidance field of AID. Only after this equivalence is established can the randomized formulation serve as the basis for continuous-time RL and actor–critic updates in Section 4.

3.1 Auxiliary Gaussian Formulation

Motivated by continuous-time relaxed-control formulations with Gaussian randomization, for each fixed observable input ξ and each $\lambda > 0$, let $\Pi^{(\lambda)}(\xi)$ denote the class of Gaussian feedback policies of the form

$$\pi^{(\lambda)}(\cdot \mid t,x;\xi) = \mathcal{N}\left(\mu^{\pi^{(\lambda)}}(t,x;\xi), \frac{2\lambda}{\beta d} I\right).$$

Thus, under $\pi^{(\lambda)}$, the control is sampled from a Gaussian policy, while the policy mean is the candidate deterministic guidance law. Unlike entropy-regularized exploratory CT-RL, we do not add an entropy term to the objective; the Gaussian randomization is introduced as a learning device whose mean is designed to remain aligned with the deterministic guidance field. Following the relaxed-control formulation of [43], the corresponding relaxed dynamics under a policy $\pi^{(\lambda)}$ are

$$\dot{x}^{\pi^{(\lambda)}}(r) = b\big(r, x^{\pi^{(\lambda)}}(r)\big) + \int_{\mathbb{R}^d} a\,\pi^{(\lambda)}(a \mid r, x^{\pi^{(\lambda)}}(r);\xi)\,\mathrm{d}a, \qquad x^{\pi^{(\lambda)}}(t) = x,$$

while the associated randomized objective is written in the centered form

$$J^{\pi^{(\lambda)}}(t,x;\xi) := \bar{\Psi}\big(x^{\pi^{(\lambda)}}(T;t,x,\xi);\xi\big) - \lambda T + \int_t^T \left(\int_{\mathbb{R}^d} \frac{\beta}{2}\|a\|^2\ \pi^{(\lambda)}(a \mid r, x^{\pi^{(\lambda)}}(r);\xi)\,\mathrm{d}a\right)\mathrm{d}r.$$

The shift $-\lambda T$ centers the auxiliary value; being policy-independent, it leaves all optimizers unchanged. Next we define

$$V^{(\lambda)}(t,x;\xi) := \inf_{\pi^{(\lambda)}\in\Pi^{(\lambda)}(\xi)} J^{\pi^{(\lambda)}}(t,x;\xi). \tag{6}$$

Under the same regularity assumptions as before, $V^{(\lambda)}$ is the unique classical solution of

$$V_t^{(\lambda)}(t,x;\xi) + \inf_{\mu\in\mathbb{R}^d}\left\{\nabla_x V^{(\lambda)}(t,x;\xi)^\top\big(b(t,x)+\mu\big) + \frac{\beta}{2}\|\mu\|^2 + \lambda\right\} = 0, \tag{7}$$

with terminal condition

$$V^{(\lambda)}(T,x;\xi) = \bar{\Psi}(x;\xi) - \lambda T. \tag{8}$$

3.2 Correspondence between Auxiliary Gaussian Problem and AID

The auxiliary Gaussian formulation is useful only if it gives rise to the (deterministic) inpainting guidance in the original problem. We first record a value-shift lemma, which aligns the deterministic and auxiliary HJB equations. We then establish the main policy correspondence in two directions: the deterministic AID optimizer is sufficient to construct an optimal auxiliary Gaussian policy, and conversely, any optimal auxiliary Gaussian policy must recover the deterministic optimizer through its mean. Detailed proofs are given in the extended version.

Lemma 3.1 (Centered value shift for the auxiliary problem)**.** *Suppose* (4)–(5) *and* (7)–(8) *admit unique classical solutions V and $V^{(\lambda)}$, respectively. Then, for every fixed observable input ξ,*

$$V^{(\lambda)}(t, x; \xi) = V(t, x; \xi) - \lambda t.$$

Lemma 3.1 shows that the auxiliary value differs from the deterministic value only by a known normalization. This value relation is an important alignment step, but by itself it does not yet justify using randomized policies for learning. The next two results establish the stronger policy-level correspondence.

Theorem 3.2 (Sufficiency: deterministic guidance induces an optimal Gaussian policy)**.** *Under the assumptions of Lemma 3.1, define*

$$u^*(t, x; \xi) := -\frac{1}{\beta} \nabla_x V(t, x; \xi).$$

Then the Gaussian policy

$$\pi^{(\lambda),*}(\cdot \mid t, x; \xi) = \mathcal{N}\left(u^*(t, x; \xi), \frac{2\lambda}{\beta d} I\right) \tag{9}$$

is optimal for the auxiliary problem (6) *within $\Pi^{(\lambda)}(\xi)$. Moreover, u^* solves the AID problem* (3)*.*

Theorem 3.2 shows that passing from the deterministic AID problem to the auxiliary Gaussian problem does not lose the desired control: the deterministic optimizer can be lifted to an optimal randomized policy by using it as the Gaussian mean. For learning, however, the reverse direction is more important. If the auxiliary problem is solved by a learning algorithm, we must also know that its mean solves the original AID problem. This is the necessity direction below.

Theorem 3.3 (Necessity: auxiliary optimality recovers deterministic guidance)**.** *Under the assumptions of Lemma 3.1, let $\pi^{(\lambda),*} \in \Pi^{(\lambda)}(\xi)$ be any optimal feedback policy for the auxiliary Gaussian problem* (6) *for every initial pair (t, x). Then its mean field $\mu^{\pi^{(\lambda),*}}(t, x; \xi)$ is the optimal deterministic guidance field for the original AID problem* (3)*; that is,*

$$\mu^{\pi^{(\lambda),*}}(t, x; \xi) = u^*(t, x; \xi).$$

Together, Lemma 3.1 and Theorems 3.2–3.3 show that the auxiliary Gaussian formulation changes the learning mechanism but not the control target. Hence the actor in Section 4 can be parameterized as the Gaussian mean and deployed deterministically.

4 Amortized Actor–Critic Algorithm

Lemma 3.1 and Theorems 3.2–3.3 identify the learning target: solving the auxiliary Gaussian problem is equivalent, at the policy-mean level, to learning the deterministic optimal guidance field u^*. This implication is essential, because policy-evaluation and policy-gradient methods in CT-RL rely on *stochastic* exploration and therefore do not apply directly to the original deterministic policies. At the same time, our auxiliary Gaussian formulation is not the entropy-regularized exploratory objective studied in [19, 20]; the Gaussian variance is fixed to preserve the deterministic optimizer rather than optimized through an entropy reward. We therefore adapt their martingale orthogonality and policy-gradient methodology to the AID-specific Bellman residual, leading to moment conditions and stochastic-approximation updates specialized to our auxiliary problem.

Parametrization. We parameterize the critic and actor by

$$\hat{V}^{\theta}(t,x;\xi) := NN^{\theta}(t,x,\xi) - \lambda t, \qquad \hat{\mu}^{\phi}(t,x;\xi) := NN^{\phi}(t,x,\xi).$$

The associated Gaussian policy is $\hat{\pi}^{(\lambda),\phi}(\cdot \mid t,x;\xi) := \mathcal{N}\Big(\hat{\mu}^{\phi}(t,x;\xi), \frac{2\lambda}{\beta d} I\Big)$.

Martingale moment conditions. Let $A_t \sim \pi^{(\lambda)}(\cdot \mid t, X_t;\xi)$, and let X_t be the state process obtained by rolling out the guided reverse dynamics under this exploratory policy during training. Thus, $\mathrm{d}\hat{V}^{\theta}(t,X_t;\xi)$ below denotes the critic increment along this same sampled-action trajectory. For the auxiliary Gaussian formulation, the quantity $\mathrm{d}V^{(\lambda)}(t,X_t;\xi) + \frac{\beta}{2}\|A_t\|^2\,\mathrm{d}t$ plays the role of the Bellman residual. Following the continuous-time actor–critic framework, we require this residual to be orthogonal to the critic and actor test functions. This yields the coupled moment conditions

$$\mathbb{E}\left[\int_0^T \partial_\theta \hat{V}^{\theta}(t,X_t;\xi)\left(\mathrm{d}\hat{V}^{\theta}(t,X_t;\xi) + \frac{\beta}{2}\|A_t\|^2\,\mathrm{d}t\right)\right] = 0,$$

$$\mathbb{E}\left[\int_0^T \partial_\phi \log\hat{\pi}^{(\lambda),\phi}(A_t \mid t,X_t;\xi)\left(\mathrm{d}\hat{V}^{\theta}(t,X_t;\xi) + \frac{\beta}{2}\|A_t\|^2\,\mathrm{d}t\right)\right] = 0,$$

where the expectation is taken over the sampled task $\zeta \sim \rho_{\mathrm{task}}$, the initial state, and the Gaussian policy randomness. The critic is anchored by the auxiliary terminal condition, implemented below through the terminal target $\hat{V}_{n,K}$. The discretized updates below are the corresponding stochastic-approximation discretizations of these two moment conditions.

Time discretization and updates. The problem is formulated in continuous time because both the diffusion dynamics and the results in Section 3 are inherently continuous-time; time discretization is introduced only for numerical implementation. We implement the method on a time grid $0 = t_0 < \cdots < t_K = T$ with $\Delta t_k := t_{k+1} - t_k$. We use Heun's method, hold the sampled action fixed across the two drift evaluations within each step, and use Euler for the final step; hence the number of function evaluations is $2K-1$ (35 for $K=18$, and 23 for $K=12$). At iteration n, we sample $\zeta_n = (M_n, y_n, x_n^{\dagger}) \sim \rho_{\mathrm{task}}$, set $\xi_n = (M_n, y_n)$, and roll out one trajectory under $\hat{\pi}^{(\lambda),\phi_n}$. Writing $\hat{V}_{n,k} := \hat{V}^{\theta_n}(t_k, X_{n,k};\xi_n)$, $\hat{\mu}_{n,k} := \hat{\mu}^{\phi_n}(t_k, X_{n,k};\xi_n)$, and $\hat{V}_{n,K} := \Psi(X_{n,K};\zeta_n) - \lambda T$, we define the one-step residual

$$\delta_{n,k} := \hat{V}_{n,k+1} - \hat{V}_{n,k} + \frac{\beta}{2}\|A_{n,k}\|^2\,\Delta t_k,$$

which is the discretized analogue of a temporal-difference residual. The updates

$$\theta_{n+1} \leftarrow \theta_n + a_n^c \sum_{k=0}^{K-1} \partial_\theta \hat{V}^{\theta_n}(t_k, X_{n,k};\xi_n)\,\delta_{n,k}, \tag{10}$$

$$\phi_{n+1} \leftarrow \phi_n - a_n^a \sum_{k=0}^{K-1} \partial_\phi \log\hat{\pi}^{(\lambda),\phi_n}(A_{n,k} \mid t_k, X_{n,k};\xi_n)\,\delta_{n,k}, \tag{11}$$

are therefore the continuous-time analogues of actor–critic updates in discrete-time RL [39, 26]. Moreover, for the Gaussian policy, $\partial_\phi \log\hat{\pi}^{(\lambda),\phi}(A \mid t,x;\xi) = \frac{\beta d}{2\lambda}\left(\partial_\phi\hat{\mu}^{\phi}(t,x;\xi)\right)^{\top}\left(A - \hat{\mu}^{\phi}(t,x;\xi)\right)$. This also shows *why* the auxiliary Gaussian formulation is necessary: the stochastic term $A - \hat{\mu}^{\phi}$ is exactly what makes the policy-gradient update nondegenerate, whereas under a deterministic policy this score term vanishes and the policy-gradient identity becomes uninformative.

Algorithm 1 summarizes the offline training and deployment procedures of AID. The amortization lies in learning shared actor and critic parameters over tasks sampled from ρ_{task}, and then reusing the learned mean guidance law on unseen masked inputs without per-instance optimization.

5 Empirical Results

Backbones and datasets. We evaluate AID within the official EDM and EDM2 pipelines [23, 25, 24], which are representative modern diffusion backbones designed for strong image quality at

Algorithm 1 Offline Training and Deployment of AID

for $n = 1, \ldots, N$ **do**
  Sample $\zeta_n = (M_n, y_n, x_n^\dagger) \sim \rho_{\text{task}}$, set $\xi_n = (M_n, y_n)$, and initialize $X_{n,0} \sim p_T$
  for $k = 0, \ldots, K-1$ **do**
    Compute $\hat{\mu}_{n,k} = \hat{\mu}^{\phi_n}(t_k, X_{n,k}; \xi_n)$ and sample $A_{n,k} \sim \mathcal{N}(\hat{\mu}_{n,k}, \frac{2\lambda}{\beta d} I)$
    Advance one Heun step, holding $A_{n,k}$ fixed within the step, to obtain $X_{n,k+1}$
  end for
  Update θ_n and ϕ_n by (10) and (11)
end for

Deployment: for a new masked input ξ, discard policy noise and use the deterministic mean guidance $\hat{\mu}^{\phi}(t_k, X_k; \xi)$ inside the same guided reverse-time solver.

low sampling budgets ($K = 18$ steps). Specifically, we use AFHQv2 [1] and FFHQ [22] with the official EDM backbones, and ImageNet [5, 35] with the official EDM2 backbone. These are the highest-resolution benchmark settings released in the corresponding official EDM/EDM2 pipelines used here, and together they let us test AID across two different pretrained diffusion families in both a pixel-space regime and a higher-resolution latent-diffusion regime. Additional mask-specific results, qualitative comparisons, parameter-overhead visualizations, and implementation details are reported in the extended version.

Baselines and scope. Our comparison set is designed to test the central deployment question: how much can be gained by amortizing a small guidance module while keeping the pretrained diffusion backbone fixed? Full inpainting networks and large learned helper modules, such as LaMa, BrushNet, and DAVI, test a different deployment regime: they amortize inference by training a model-scale component, whereas our goal is to isolate the cost–quality trade-off of a tiny reusable guidance module on a fixed backbone. We compare against representative fixed-model inpainting baselines spanning several major adaptation directions: Unguided, Replacement [38], MCG [2], DPS [3], and RePaint [31]. These baselines cover no conditioning, replacement-based heuristics, manifold-constrained guidance, posterior-gradient guidance, and iterative resampling respectively. In particular, RePaint is a strong and widely used benchmark in this line of work. For the latent EDM2 setting, we also include LatentPaint [4], the closest reusable-module competitor available in our latent-backbone setting. We therefore evaluate AID against both strong fixed-model methods and the most relevant latent-space amortized baseline. We report masked-region PSNR together with SSIM [44] and LPIPS [45] on the completed image, using two AID operating points: the default setting $K = 18$ (AID-18) for best quality and the low-latency setting $K = 12$ (AID-12) for faster sampling. Importantly, the low-latency setting $K = 12$ is obtained by directly deploying the same guidance module trained for the default $K = 18$ setting, without retraining.

5.1 Pixel-Space EDM Benchmarks: AFHQv2 and FFHQ

We first evaluate on AFHQv2 and FFHQ, the two pixel-space EDM image benchmarks. The main free-form results are summarized in Table 1, the complete frontier is shown in Figure 1, and a visual comparison is presented in Figure 2. On both datasets, AID-18 achieves the strongest performance on every free-form metric among all compared methods while using the same NFE as the standard 35-NFE baselines and only one tenth of RePaint's sampling budget. The improvement is consistent across fidelity-oriented metrics and perceptual quality, indicating that the learned guidance does not merely improve visible-region matching but also improves the overall completed image. The margins over Replacement, MCG, and DPS are larger still, suggesting that the gain cannot be explained by simple overwrite, projection, or posterior-guidance heuristics.

This advantage is not confined to the main free-form benchmark. AID-18 remains best on all metrics for both center and strip masks on AFHQv2 and FFHQ, implying that the improvement is stable across unseen mask families rather than concentrated on a single setting. The low-latency operating point serves a different purpose: without any retraining, it substantially reduces inference time while still outperforming Unguided, Replacement, MCG, and DPS across these additional mask settings, and remains competitive with RePaint at only $\frac{1}{15}$ of its sampling budget. The same learned guidance

Table 1: Free-form inpainting results on 1000 test images. Entries report mean ± standard deviation over three seeds; LatentPaint is included only for ImageNet.

Data	Method	NFE	Sec.	PSNR ↑	SSIM ↑	LPIPS ↓
AFHQv2						
	Unguided	35	1.54	4.85 ± 0.06	0.4296 ± 0.0035	0.2716 ± 0.0025
	Replace	35	1.55	7.15 ± 0.08	0.5164 ± 0.0052	0.1931 ± 0.0026
	MCG	35	1.58	7.08 ± 0.02	0.5172 ± 0.0053	0.1915 ± 0.0015
	DPS	35	2.41	10.00 ± 0.02	0.6336 ± 0.0048	0.1257 ± 0.0019
	RePaint	350	15.50	10.97 ± 0.03	0.6816 ± 0.0031	0.0879 ± 0.0006
	AID-12	**23**	**1.03**	11.09 ± 0.04	0.6815 ± 0.0035	0.0955 ± 0.0017
	AID-18	35	1.57	**13.01 ± 0.09**	**0.7336 ± 0.0034**	**0.0772 ± 0.0015**
FFHQ						
	Unguided	35	1.55	4.65 ± 0.00	0.4607 ± 0.0042	0.2152 ± 0.0030
	Replace	35	1.56	6.93 ± 0.08	0.5585 ± 0.0050	0.1532 ± 0.0015
	MCG	35	1.59	6.92 ± 0.04	0.5633 ± 0.0049	0.1507 ± 0.0022
	DPS	35	2.40	10.41 ± 0.03	0.7028 ± 0.0027	0.0792 ± 0.0011
	RePaint	350	15.50	10.88 ± 0.05	0.7276 ± 0.0031	0.0677 ± 0.0013
	AID-12	**23**	**1.04**	11.96 ± 0.04	0.7306 ± 0.0025	0.0662 ± 0.0011
	AID-18	35	1.58	**12.87 ± 0.06**	**0.7598 ± 0.0027**	**0.0597 ± 0.0009**
ImageNet						
	Unguided	35	1.84	3.53 ± 0.07	0.4311 ± 0.0025	0.5084 ± 0.0025
	Replace	35	1.94	5.34 ± 0.02	0.4669 ± 0.0025	0.4275 ± 0.0028
	MCG	35	1.86	5.29 ± 0.08	0.4692 ± 0.0018	0.4242 ± 0.0018
	DPS	35	2.50	7.26 ± 0.07	0.5010 ± 0.0024	0.4057 ± 0.0019
	LatentPaint	35	2.01	6.47 ± 0.06	0.4874 ± 0.0025	0.3989 ± 0.0028
	RePaint	350	19.06	7.56 ± 0.03	0.5204 ± 0.0019	0.3272 ± 0.0026
	AID-12	**23**	**1.22**	9.67 ± 0.04	0.5491 ± 0.0092	0.3321 ± 0.0009
	AID-18	35	1.92	**9.79 ± 0.04**	**0.5498 ± 0.0088**	**0.3108 ± 0.0006**

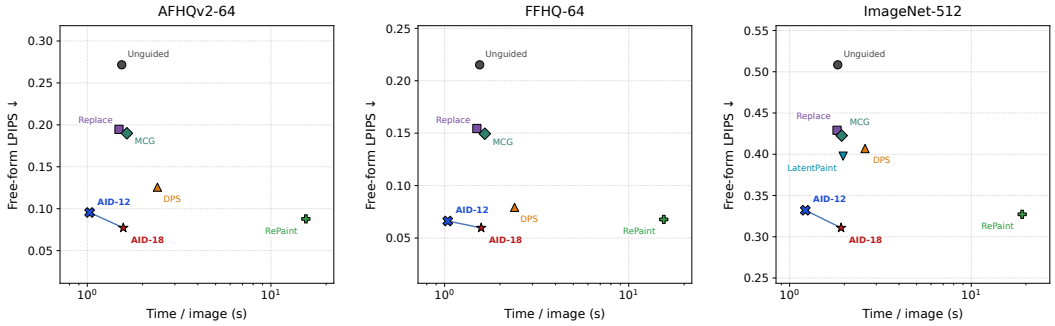


Figure 1: Quality–speed frontier on free-form inpainting. Each panel corresponds to one dataset; ImageNet also includes LatentPaint. Lower LPIPS and lower wall-clock time are better.

module can therefore support both a best-quality regime and a low-latency regime without reverting to per-instance optimization.

5.2 Latent-Space EDM2 Benchmark: ImageNet

We next evaluate on ImageNet using the pretrained EDM2 latent backbone. This setting is more demanding because the images have higher resolution 512×512, and sampling is performed in latent space whereas the final outputs are decoded by a VAE and evaluated in pixel space. It also provides the most relevant comparison to LatentPaint. As shown in Table 1, on the free-form benchmark, AID-18 improves over RePaint on all three metrics while using only about one tenth as many function

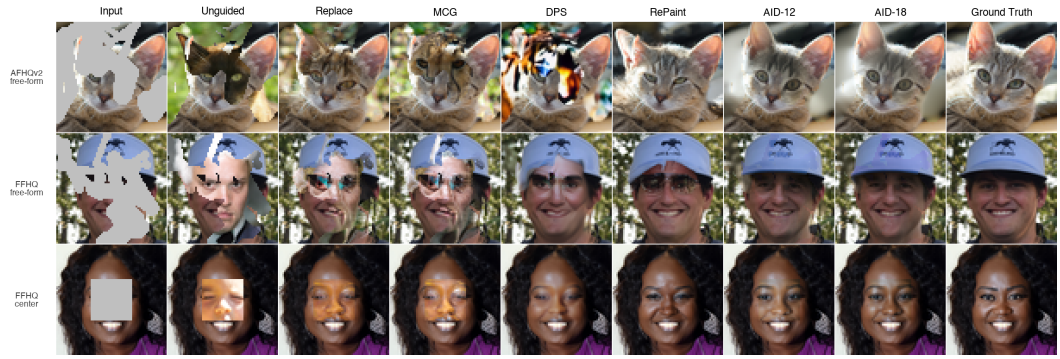


Figure 2: Visual comparisons for pixel-space EDM.

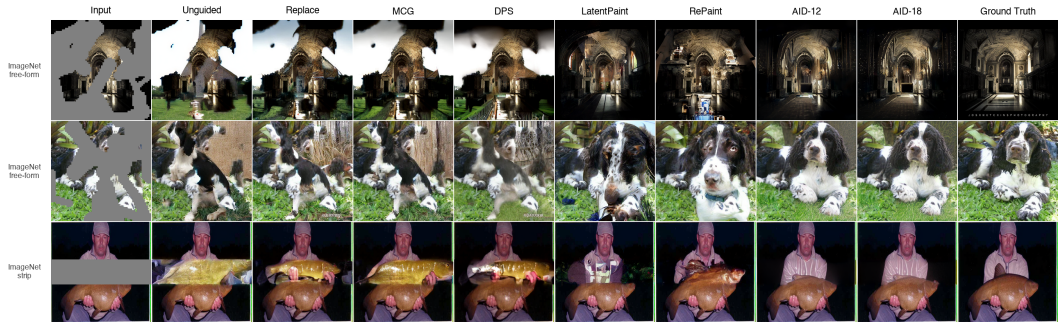


Figure 3: Visual comparisons for latent-space EDM2.

evaluations. It also outperforms LatentPaint on every reported metric, despite using a guidance module with less than 45% of the trainable parameters of LatentPaint's propagation module. See also a visualization in Figure 3.

The higher-resolution setting also sharpens the speed-quality story. The low-latency AID-12 operating point already surpasses all non-RePaint baselines and LatentPaint on every metric with significant margins, and it also improves over RePaint on PSNR and SSIM while giving up only a small amount on LPIPS. Across unseen center and strip masks, the same pattern remains visible: the default point is the strongest overall configuration, while the low-latency point still improves substantially over the non-RePaint baselines and LatentPaint, and remains competitive with RePaint. The same amortized-guidance principle therefore remains effective when the pretrained model is replaced by the substantially more demanding latent EDM2 backbone at 512×512.

5.3 Efficiency and Parameter Overhead

The quality gains above are accompanied by a favorable deployment profile. The trainable module is tiny relative to the pretrained score model: 409K parameters on EDM (0.66% of the score network) and 411K on EDM2 (0.33%). On ImageNet this is also less than 45% the size of LatentPaint's learned propagation module, while giving better results on every reported metric and mask type. Within each dataset/backbone, the $K = 18$ operating point has essentially the same wall-clock cost as the 35-NFE baselines, but delivers markedly better quality. Compared with RePaint, it provides roughly a $10\times$ speedup across all three datasets. The $K = 12$ operating point pushes further along the same frontier and reaches $15\times$ speedup over RePaint while remaining competitive in quality. Taken together, these results support the main empirical claim of the paper: amortizing the guidance module moves up the entire quality–speed frontier of pretrained diffusion models, rather than merely recovering a different point on the same curve.

6 Conclusion

In this paper we introduce AID, a continuous-time actor–critic framework for image inpainting that keeps the diffusion backbone fixed and learns a small reusable guidance module offline. We formulate amortized inpainting as a deterministic guidance control problem and establish a policy-equivalence bridge to an auxiliary Gaussian formulation, preserving the deterministic guidance field used at deployment. Empirically, across AFHQv2, FFHQ, and ImageNet under the pixel EDM and latent EDM2 pipelines, AID improves the quality–speed trade-off over strong fixed-backbone and amortized inpainting baselines while adding less than one percent trainable overhead.

More broadly, this work adds to a recent line of research suggesting that control- and RL-based frameworks can provide a theoretically grounded way to study and improve diffusion models. While this direction is still in its early innings, our results indicate that continuous-time RL underpins not only rigorous analysis but also practical algorithm design in diffusion-based generation. We hope AID can motivate further work on controlled guidance, amortized adaptation, and broader uses of continuous-time RL for generative AI.

Acknowledgments and Disclosure of Funding

Funding. Yilie Huang acknowledges financial support from the Start-up Fund of The Hong Kong Polytechnic University (Project ID: P0063874). Xun Yu Zhou is supported by the Nie Center for Intelligent Asset Management and the Yao Artificial Intelligence Research Fund at Columbia University.

Competing Interests. The authors declare no competing interests.

References

[1] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8188–8197, 2020.

[2] Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. In *Advances in Neural Information Processing Systems*, volume 35, pages 25683–25696, 2022.

[3] Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023.

[4] Ciprian Corneanu, Raghudeep Gadde, and Aleix M Martinez. LatentPaint: Image inpainting in latent space with diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4334–4343, 2024.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

[6] Alexander Denker, Francisco Vargas, Shreyas Padhy, Kieran Didi, Simon Mathis, Vincent Dutordoir, Riccardo Barbano, Emile Mathieu, Urszula Julia Komorowska, and Pietro Lio. DEFT: Efficient fine-tuning of diffusion models by learning the generalised h-transform. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

[7] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794, 2021.

[8] Xuefeng Gao, Jiale Zha, and Xun Yu Zhou. Reward-directed score-based diffusion models via q-learning. *Journal of Machine Learning Research*, 26(302):1–46, 2025.

[9] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020.

[11] Yilie Huang. Continuous-time reinforcement learning for asset–liability management. In *Proceedings of the ACM International Conference on AI in Finance*, pages 360–368, 2025.

[12] Yilie Huang and Xun Yu Zhou. Data-driven exploration for a class of continuous-time indefinite linear–quadratic reinforcement learning problems. *IEEE Transactions on Automatic Control*, 2026. Early Access; doi: 10.1109/TAC.2026.3734857.

[13] Yilie Huang and Xun Yu Zhou. Image AID via continuous-time reinforcement learning, 2026. Extended version; arXiv:2605.13010.

[14] Yilie Huang, Yanwei Jia, and Xun Yu Zhou. Achieving mean–variance efficiency by continuous-time reinforcement learning. In *Proceedings of the ACM International Conference on AI in Finance*, pages 377–385, 2022.

[15] Yilie Huang, Yanwei Jia, and Xun Yu Zhou. Sublinear regret for a class of continuous-time linear-quadratic reinforcement learning problems. *SIAM Journal on Control and Optimization*, 63(5):3452–3474, 2025.

[16] Yilie Huang, Yanwei Jia, and Xun Yu Zhou. Mean–variance portfolio selection by continuous-time reinforcement learning: Algorithms, regret analysis, and empirical study. *Management Science*, 2026. To appear; arXiv:2412.16175.

[17] Yilie Huang, Wenpin Tang, and Xun Yu Zhou. ART for diffusion sampling: A reinforcement learning approach to timestep scheduling. In *Advances in Neural Information Processing Systems*, 2026. To appear; arXiv:2601.18681.

[18] Yilie Huang, Wenpin Tang, and Xun Yu Zhou. Adaptive reparametrized time for score-based diffusion sampling. *Journal of Machine Learning Research*, 2026. To appear; arXiv:2607.02137.

[19] Yanwei Jia and Xun Yu Zhou. Policy evaluation and temporal-difference learning in continuous time and space: A martingale approach. *Journal of Machine Learning Research*, 23(154):1–55, 2022.

[20] Yanwei Jia and Xun Yu Zhou. Policy gradient and actor-critic learning in continuous time and space: Theory and algorithms. *Journal of Machine Learning Research*, 23(275):1–50, 2022.

[21] Xuan Ju, Xian Liu, Xintao Wang, Yuxuan Bian, Ying Shan, and Qiang Xu. BrushNet: A plug-and-play image inpainting model with decomposed dual-branch diffusion. In *European Conference on Computer Vision*, pages 150–168. Springer, 2024.

[22] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[23] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, pages 26565–26577, 2022.

[24] Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. In *Advances in Neural Information Processing Systems*, volume 37, pages 52996–53021, 2024.

[25] Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24174–24184, 2024.

[26] Vijay Konda and John Tsitsiklis. Actor-critic algorithms. In *Advances in Neural Information Processing Systems*, volume 12, 1999.

[27] Sojin Lee, Dogyun Park, Inho Kong, and Hyunwoo J Kim. Diffusion prior-based amortized variational inference for noisy inverse problems. In *European Conference on Computer Vision*, pages 288–304. Springer, 2024.

[28] Henry Li and Marcus Pereira. Solving inverse problems via diffusion optimal control. In *Advances in Neural Information Processing Systems*, volume 37, pages 73549–73571, 2024.

[29] Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. MAT: Mask-aware transformer for large hole image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10758–10768, 2022.

[30] Kendong Liu, Zhiyu Zhu, Chuanhao Li, Hui Liu, Huanqiang Zeng, and Junhui Hou. PrefPaint: Aligning image inpainting diffusion model with human preference. In *Advances in Neural Information Processing Systems*, volume 37, pages 30554–30589, 2024.

[31] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. RePaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022.

[32] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

[33] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings of the International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[34] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[35] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[36] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022.

[37] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[38] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[39] Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, 1988.

[40] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with Fourier convolutions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2149–2159, 2022.

[41] Wenpin Tang and Xun Yu Zhou. Regret of exploratory policy improvement and q-learning. *arXiv preprint arXiv:2411.01302*, 2024.

[42] Masatoshi Uehara, Yulai Zhao, Kevin Black, Ehsan Hajiramezanali, Gabriele Scalia, Nathaniel Lee Diamant, Alex M. Tseng, Tommaso Biancalani, and Sergey Levine. Fine-tuning of continuous-time diffusion models as entropy-regularized control. *arXiv preprint arXiv:2402.15194*, 2024.

[43] Haoran Wang, Thaleia Zariphopoulou, and Xun Yu Zhou. Reinforcement learning in continuous time and space: A stochastic control approach. *Journal of Machine Learning Research*, 21 (198):1–34, 2020.

[44] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

[45] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

[46] Hanyang Zhao, Haoxian Chen, Ji Zhang, David Yao, and Wenpin Tang. Score as Action: Fine tuning diffusion generative models by continuous-time reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 77371–77389. PMLR, 2025.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the frozen-backbone, amortized-guidance, theoretical-bridge, and empirical quality–speed claims. These claims are supported by Sections 2–3, Section 4, and Section 5.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper delimits its claims through the frozen-backbone deployment regime, comparison scope, datasets, mask protocols, and sampling budgets described in Section 5.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: The mathematical setup and theorem statements are given in Sections 2–3. Complete proofs of the value-shift and Gaussian-policy recovery results are provided in the extended version.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: Section 5 specifies datasets, pretrained backbones, baselines, metrics, sampling budgets, and repeated-run reporting; complete hyperparameters and training/evaluation protocols are provided in the extended version and released code.

Guidelines:

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

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

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

   Answer: [Yes]

   Justification: The experiments use public benchmark datasets and public pretrained EDM/EDM2 backbones. An anonymized supplementary package provides the implementation, configuration files, and reproduction instructions for AID and the evaluated baselines.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 5 describes the evaluation protocol and baselines; the extended version gives the architecture, pretrained backbones, solver/time grid, task distribution, optimizer, and hyperparameters.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The quantitative tables report mean and standard deviation over three random seeds for the main metrics, and the repeated-run protocol is stated in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Section 5 reports sampling-time measurements and trainable parameter overhead. The extended version and released configurations provide optimizer/training settings, approximate guidance-training cost, and per-dataset details.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The work uses public benchmark datasets and pretrained models for image inpainting research, without human-subject experiments, new data collection, or privacy-sensitive deployment studies.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [N/A]

Justification: The paper is a methodological study evaluated on standard inpainting benchmarks and does not deploy a user-facing editing system, collect new human-subject data, or release a new generative backbone. Potential misuse follows the general risks of existing image-editing and inpainting systems rather than a new risk introduced by this work.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [N/A]

Justification: The paper does not release a new scraped dataset, a new foundation image generator, or a user-facing image-editing service. It releases a lightweight inpainting guidance method built on existing public pretrained backbones.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: The paper cites the datasets, pretrained backbones, baseline methods, and code assets used in the experiments. The supplementary material includes an asset summary with corresponding sources, licenses, and usage terms, and our use follows these terms.

Guidelines:

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

13. **New assets**

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

Answer: [Yes]

Justification: The anonymized supplementary package documents the AID implementation, configuration files, and reproduction scripts. The paper does not introduce a new dataset.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The work does not involve crowdsourcing or human-subject experiments.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The work does not involve human-subject data collection or human-subject experiments.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: The core method, theory, experiments, and evaluations do not use LLMs as an important, original, or non-standard component.

Guidelines:

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