Rotations on Latent Hyperspheres: A Geometry-Aware Guiding Framework for Diffusion Models

Luca Sacchetto
School of Computation,
Information and Technology
Technical University of Munich
luca.sacchetto@tum.de

Klaus Diepold
School of Computation,
Information and Technology
Technical University of Munich
kldi@tum.de

Abstract

Diffusion models have emerged as a powerful tool across diverse domains. However, their purely data-driven nature can produce samples that violate domain-governing constraints. We introduce a plug-and-play reinforcement learning framework that optimizes initial noise samples in the latent space of frozen, pre-trained diffusion models. Leveraging the near-spherical geometry of high-dimensional Gaussian distributions, we introduce a novel rotation-matrix-based scheme for efficient latent space exploration. This steers the model toward more feature-preserving outputs, guided by task-specific rewards. We evaluate our method on three diffusion models: one trained on solutions of the Darcy Flow PDE, one on a synthetic dataset with complex structural features, and a text-conditioned one. Across all three settings, our framework yields significant improvements in sample quality, achieving a $\sim 27\%$ relative reduction in PDE residual, up to a $\sim 41\%$ relative improvement on the synthetic dataset's feature-alignment metric, and substantial gains in ImageReward, compared to the vanilla diffusion models. Finally, we show that rotation-matrix-based exploration significantly outperforms unconstrained exploration, validating our geometry-aware approach and establishing a more effective method for latent space control.

1 Introduction

Deep generative models, particularly diffusion [1] and latent diffusion models (LDM) [2], have emerged as remarkably powerful tools for learning complex data distributions. Their success has been more prominent in image synthesis, where they have revolutionized computer vision and content creation. However, their applicability extends far beyond the visual arts, with successful deployments in scientific and engineering domains such as 3D modeling [3], audio synthesis [4–6], molecular generation [7–11], protein design [12–14], physics simulation [15–17] and material design [18].

Despite their impressive capabilities, a key limitation of these purely data-driven models is their tendency to produce samples that may not adhere to known, domain-governing constraints. This is particularly critical in scientific applications where outputs must satisfy physical laws, mathematical principles, or structural requirements. Furthermore, many applications benefit from steering the generative process towards a specific, desirable region of the output distribution for downstream tasks. Prominent examples include generating solutions to partial differential equations (PDEs), where outputs must remain consistent with physical principles; designing 3D models with strict geometric tolerances; or creating novel proteins that adhere to fundamental biochemical constraints.

To address this, several techniques for guiding generative models have been proposed. These range from costly fine-tuning the model's weights or incorporating constraints directly into the training objective to latent space optimization (LSO). LSO has emerged as a flexible, post hoc alternative

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

that operates on the initial noise vectors of a pre-trained, frozen generative model, often guided by reward signals to align outputs with specific objectives, such as human preferences. Nevertheless, within the current LSO landscape, we identify two significant gaps. First, most existing methods are computationally expensive, requiring iterative optimization and multiple full denoising passes to generate a single sample, which creates a substantial bottleneck at inference time. Second, current approaches often treat the latent space as a generic Euclidean space, overlooking the near-spherical geometry of high-dimensional Gaussian noise. We posit that failing to account for this inherent structure leads to inefficient exploration and can degrade sample quality.

In this work, we propose a novel framework to address these shortcomings. Our main contributions are:

- We introduce Rotational Latent Space Optimization (RLSO), a novel, geometry-aware, modular exploration technique for the latent space of frozen diffusion models. RLSO leverages rotation matrices to preserve the norm of latent vectors, respecting the inherent geometry of the Gaussian prior.
- We formulate the optimization of the initial noise as a reinforcement learning (RL) problem, allowing us to amortize the optimization cost. This enables the generation of constraint-aligned samples with only a single denoising pass at inference time, drastically improving efficiency.
- We provide a validation of our approach on three distinct use cases from vastly different domains, demonstrating its versatility and effectiveness in improving sample quality.

Our experiments show that our geometry-aware RLSO framework significantly outperforms standard, unconstrained exploration. Furthermore, we demonstrate that the modularity of our approach can be leveraged to control the trade-off between generalization and computational efficiency, establishing a new and effective paradigm for latent space control in diffusion models.

2 Related Work

High-Dimensional Rotations The study of the theory and application of rotations in dimensions higher than three originates in the 18th century, but the literature remains limited and lacks a unified taxonomy. While some applications have successfully used Rodrigues' formula [19] or Cayley's transform [20, 21], for our work we use rotation matrices as a representation of rotations. Specifically, we follow Schoute [22]'s theoretical generalization of Euler [23]'s *Principal Rotation Theorem* to n dimensions:

Any displacement of a rigid body about a fixed point in n dimensions can be achieved for n even by $\frac{n}{2}$ simple rotations in mutually orthogonal planes about the fixed point and for n odd by $\frac{n-1}{2}$ such rotations. Furthermore the rotations commute.

In practice, we rely on Mortari [24]: by using the properties of the eigenanalysis of rotation matrices and an n-dimensional extension of the vector cross product [25], the author provides a formula to construct simple rotations. This formulation allows us to uniquely identify a rotation matrix by defining a rotation angle and a principal plane of rotation.

Latent Space of Diffusion Models We define the latent space of a diffusion model as the Gaussian space $(\mathcal{X}, \mathcal{N}(0, I_{chw}))$ containing all possible initial noise samples $x \in \mathbb{R}^{c \times h \times w}$, where x is a realization of $X \sim \mathcal{N}(0, I_{chw})$, c is the number of channels, h is the height, and w is the width of the latent samples. In high dimensions, due to the *Concentration of Measure Phenomenon* [26] and the *Gaussian Annulus Theorem* [27], Gaussian samples have norms concentrated around the square root of the dimension n, i.e., they lie in a thin shell of a hypersphere with radius $\sqrt{n}$. This means that the latent space behaves more akin to a hyperspherical space than a Euclidean one.

Though underexplored, this phenomenon has been described [28, 29] and exploited, either by exploring the latent space with norm-regularization techniques or spherical linear interpolation [30–33]. The works of Park et al. [34] and Jin et al. [35] advance in this direction and use geodesic shooting for latent space exploration and Rodrigues' formula-based rotation for guidance, respectively. We note that, while our work focuses on diffusion models specifically, this interpretation is valid for any kind of generative model that possesses a Gaussian latent space.

Latent Space Optimization Latent space optimization, or noise optimization, is a rapidly growing field that encompasses all algorithms and techniques which aim to optimize the input noise (or some intermediary latent space) of frozen, pre-trained generative models. A common approach is to optimize latent samples with classic optimization techniques, either by selecting the best candidate out of a population [36] or by backpropagating the gradients through the full denoising process [31, 37–39]. Alternatively, Eyring et al. [40] use a one-step diffusion model. These do achieve significant improvements, but have two significant drawbacks: backpropagating the gradients through the denoising process may be costly, and importantly, they add significant computational overhead at inference time, because they need to run the denoising process multiple times [41].
On the other hand, this issue can also be mitigated by training an auxiliary model to optimize the latent samples, so that only one pass through the auxiliary model and the denoising process is required at inference time. Lu et al. [42] use an auxiliary model to predict the values of an energy function to guide the sampling process, Ahn et al. [43] train a model in the latent space to mimic classifier-free guidance, while Venkatraman et al. [44] train a model to substitute the sampler by learning the reverse denoising process of high-reward samples. Most similarly to our work, Eyring et al. [45] recently proposed training a Low-Rank Adaptation (LoRA) network to predict latent samples that denoise into high-reward samples. Wagenmaker et al. [46] also use RL to steer the latent-noise space of a frozen diffusion policy, though in the distinct setting of downstream robot-control objectives. While these papers present similarities to our work, namely training auxiliary models to generate optimized latent samples for pre-trained diffusion models, our approach introduces several key differences. Typically, the techniques employed to ensure that optimized latent samples remain Gaussian and within the *shell* range from adding a penalty term to the cost function [40, 45, 44], adding a small Gaussian perturbation to each update step [39], projecting back to the shell and adding small Gaussian perturbations [37], or using small, regularized gradient steps [39]. Instead of relying on penalizing terms with high overhead or projecting back to the shell, which limits exploration, our rotation-matrix-based exploration strategy offers a principled and geometry-aware method for generating optimized latent samples that automatically remain both on the shell and semantically linked to the original latent sample. Moreover, this exploration strategy allows for modular control over the direction and angle. In contrast to Ahn et al. [43] and Eyring et al. [45], our use of policy gradient training enables compatibility with arbitrary reward functions, even when their gradients are intractable. Compared to Venkatraman et al. [44], who train a large U-Net diffusion model, our approach is far more lightweight, requiring an order of magnitude fewer parameters.

Approaches whose noise optimization is an inherent part of training the main model naturally do not suffer from this issue, such as Hu et al. [47], who train the encoder of their encoder-decoder structure as an RL policy in the latent space. Finally, Zhang et al. [48] expand on the work of Lu et al. [42] by integrating the energy-function guidance in the training of the main model.

3 Methodology

Our approach employs an RL agent to navigate the latent space of a frozen, pre-trained LDM. Intuitively, a policy network is trained to modify the latent vectors before denoising begins. During training, the policy proposes one or more rotations of the latent vector. Then, the rotated vector is decoded by the frozen diffusion model, and the resulting samples are scored by a task-specific reward function. Finally, the policy network is updated so that future rotations are more likely to generate high-reward samples. At inference time and in each episode step, the agent observes the current latent vector, chooses a rotation direction and angle, and applies this rotation to produce a new latent vector. Algorithm 1 describes the procedure in detail. Importantly, while training requires multiple denoising passes in order to update the policy, once the policy is learned, inference only requires a few inexpensive passes through the policy and a single denoising pass of the LDM.

3.1 Rotation Matrix

Since our intention is not to find an optimal rotation over all of $SO(n)$, but rather to use the rotation as an exploration strategy to find an optimal latent vector, the full parameterization is not required. Intuitively, the aim is to find an optimal point, not an optimal path to it. Furthermore, because parameter efficiency is paramount in machine learning algorithms, the full parameterization can be deleterious. Therefore, we define a parameter description of n-dimensional rotations that can encompass all mathematical properties of a general rotation and offer modularity for managing the

Algorithm 1 RLSO

Input: Pre-trained LDM $D(\cdot)$, reward function $r(\cdot)$

Training

Initialize: Policy network $\pi_\phi(a|s)$ with parameters ϕ.
for each training iteration **do**
  Sample initial latent vector $s_0 \sim \mathcal{N}(0, I)$.
  for $t = 0, \ldots, T-1$ **do** ▷ Loop over steps in an episode
    Sample action $a_t \sim \pi_\phi(\cdot|s_t)$.
    Construct matrix $R(a_t)$ and compute the new latent vector: $s_{t+1} = R(a_t)s_t$.
    Generate sample by decoding the latent vector: $y_{t+1} = D(s_{t+1})$.
    Compute reward: $r_{t+1} \leftarrow r(y_{t+1})$.
    if episode terminates **then**
      break
    end if
  end for
  Update policy ϕ using PPO with the collected trajectory.
end for

Inference

Input: Trained policy π_{ϕ^*}, initial latent vector $s_0 \sim \mathcal{N}(0, I)$.
for $t = 0, \ldots, T-1$ **do**
  Sample action $a_t \sim \pi_{\phi^*}(\cdot|s_t)$.
  Construct matrix $R(a_t)$ and compute the new latent vector: $s_{t+1} = R(a_t)s_t$.
end for
Set $s_{\text{opt}} = s_T$
Generate the final, optimized sample: $y_{opt} = D(s_{opt})$.
return y_{opt}.

trade-off between generalization capabilities and computational efficiency. To this end, we adopt the theoretical framework described in Section 2, which posits that a general rotation in even dimensions can be described by $\frac{n}{2}$ mutually orthogonal planes (the principal rotation planes) and corresponding $\frac{n}{2}$ angles. Based on this framework, Lounesto [49] identifies three special classes of rotations. These are simple rotations, in which only one principal plane has a nonzero angle; double rotations, in which exactly two principal planes have nonzero angles; and isoclinic rotations, a special case of double rotations in which the two nonzero angles are equal. We extend this classification system to higher dimensions and introduce the c-fold rotation. For even n, a general rotation operates in $\frac{n}{2}$ principal planes of rotation. If exactly $c \leq \frac{n}{2}$ of these mutually orthogonal planes have nonzero rotation angles, we call the transformation a c-fold rotation. In his paper, Mortari [24, Eq. 18] provides a formula for a rotation matrix that describes a simple rotation as a function of the rotation angle and the vectors spanning the plane of rotation:

$$R(P, \theta) = I_n + (\cos\theta - 1)PP^T + P \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} P^T \sin\theta, \tag{1}$$

where θ is the angle of rotation and $P = [p_1 \quad p_2] \in \mathbb{R}^{n\times 2}$ is a matrix whose columns form an orthonormal basis for the plane of rotation. Using the fact that a general rotation can be expressed as a product of $\frac{n}{2}$ simple rotations in mutually orthogonal planes, we extend Equation 1 for rotation matrices of general rotations.

$$R_{\text{gen}} = \prod_{i=1}^{\frac{n}{2}} R(P_i, \theta_i), \tag{2}$$

where $\begin{bmatrix} P_1 & P_2 & \ldots & P_{\frac{n}{2}} \end{bmatrix} \in \mathbb{R}^{n\times n}$ is an orthonormal basis of $\mathbb{R}^n$ and $\begin{bmatrix} \theta_1 & \theta_2 & \ldots & \theta_{\frac{n}{2}} \end{bmatrix}$ are the rotation angles. We note that Equation 2 simplifies to Equation 1 for only one $\theta_i \neq 0$.

We apply these rotations to latent samples by interpreting each channel as a point in an n-dimensional space. Therefore, the channels are flattened such that they live in $\mathbb{R}^n$, where $n = hw$. The same rotation matrix $R \in \mathbb{R}^{n\times n}$ is then applied to each channel separately.

3.1.1 Simple Vector Rotation

In the case of a single-channel latent vector, we can use a simple rotation to rotate the latent vector towards a direction v. Let x be a point on the surface of a sphere centered at the origin, and let v be a unit direction vector in the tangent space of the sphere at x. Then x can be rotated towards v by angle θ with Equation 1:

$$x_{\text{rot}} = R(P, \theta)x, \qquad P \coloneqq [\hat{x} \quad v], \qquad \hat{x} = \frac{x}{\|x\|_2}, \tag{3}$$

3.1.2 c-fold Vector Rotation

In the context of a multi-channel latent space, a c-fold rotation can be used to rotate the c channels simultaneously. Let $x = [x_1, \ldots, x_c]$ be c points, each on the surface of a sphere centered at the origin. Let $[v_1, \ldots, v_c]$ be unit direction vectors in the tangent spaces at $[x_1, \ldots, x_c]$. Then they can be transformed with a c-fold rotation by angles $\theta_1, \ldots, \theta_c$ using Equation 2:

$$x_{\text{rot}} = \left(\prod_{i=1}^{c} R(P_i, \theta_i) \right) x, \qquad P_i \coloneqq \begin{cases} [\hat{x}_i \; v_i^{\perp}], & i = 1, \\ [\hat{x}_i^{\perp} \; v_i^{\perp}], & i > 1, \end{cases} \tag{4}$$

with

$$(\hat{x}_1, v_1^{\perp}, \hat{x}_2^{\perp}, v_2^{\perp}, \ldots, \hat{x}_c^{\perp}, v_c^{\perp}) = \text{GS}(\hat{x}_1, v_1, \hat{x}_2, v_2, \ldots, \hat{x}_c, v_c),$$

where $\text{GS}(\cdot)$ denotes the Gram-Schmidt orthogonalization [50] and $\hat{x_i} = \frac{x_i}{\|x_i\|_2}$.

This has two key advantages: Firstly, we can describe a rotation with $c(n+1)$ scalar parameters, which, for $c \ll \frac{n}{2}$, is far fewer than are needed to represent a general rotation. Secondly, because random vectors in high dimensions are almost orthogonal with high probability [51], this construction helps prevent the optimization from exploring rotations that have little to no effect on the vector's position. This formulation provides significant modularity. Unlike common exploration techniques such as back-projection, our approach decouples the angle and direction of rotation. This allows for fine-grained control over the scope and nature of the directional exploration by enabling precise manipulation of the number and angles of rotation planes.

3.2 Reinforcement Learning

Our aim is to optimize latent vectors with respect to some task-specific evaluator on output samples. Instead of optimizing the latents directly, our method uses RL to optimize the parameters of a policy that induces a mapping from latent vectors to optimized latent vectors. To induce this mapping, the policy defines a rollout in latent space, i.e., a series of T updates to the initial latent vector. The general RL problem is defined as a Markov decision process (MDP) characterized by the tuple (S, A, P, r, γ), where S is the state space, A is the action space, $P(s'|s, a)$ is the system transition probability, $r(s, a)$ is the reward, and $\gamma \in (0, 1)$ is the discount factor. The goal of the RL algorithm is to find an optimal policy $\pi^*(a|s)$ that maximizes the expected cumulative discounted reward. In the following paragraphs, we define the MDP tuple specific to our optimization problem.

Observation Space (S) The observation space is equal to the current latent vector in the optimization process. At step $t = 0$, the agent receives observation

$$s_0 = \text{vec}(x) \in \mathbb{R}^{n \times c} \sim \mathcal{N}(0, I), \quad n = wh,$$

which is a latent sample with its spatial dimensions flattened, where w and h are the spatial dimensions, and c is the number of channels. At any subsequent optimization step t, the agent receives the latent vector resulting from the previous state transition.

Reward Function (r) The reward function is task-specific and assumes the existence of an external evaluation function $\varepsilon(\cdot)$ that computes a scalar evaluation signal on a decoded data sample. The reward function is then built as a linear transformation of ε:

$$r(D(x)) = M\varepsilon(D(x)) + C,$$

where $D(\cdot)$ is a frozen diffusion model, x is a latent vector, and M and C are constants that ensure that most of $r(x)$ is negative. This is important for two reasons: firstly, to encourage the agent to find

the shortest path, i.e., the smallest number of optimization steps, to an optimal sample. Secondly, it discourages the agent from lingering in moderately good states to accumulate positive rewards. During training, an episode terminates either upon reaching a predefined reward threshold τ or after 15 steps. Setting the reward threshold is a heuristic process that involves calibrating a trade-off between optimization pressure and learning efficiency. A threshold that is too high may make terminal rewards too rare for efficient learning. In practice, we found that the reward value exceeded by the top 5% of samples generated by the LDM was a reasonable trade-off.

To train the agent, we employ proximal policy optimization (PPO) [52], a state-of-the-art on-policy policy optimization algorithm. While the goal of the policy is not necessarily to find trajectories to optimal samples, but rather to identify them in one or a few steps, a contextual bandit formulation would be mathematically sufficient. The choice to instead formulate it as a multi-step MDP and optimize the policy with PPO is empirically motivated: we evaluate its benefits using a controlled one-step baseline in Appendix A.3.

System Transition Probability (P) The policy $\pi_\phi(a_t|s_t)$ defines a rollout in latent space through the state-update function f:

$$s_{t+1} = f(s_t, a_t),$$

which induces the transition

$$P(s_{t+1}|s_t, a_t) = \delta(s_{t+1} - f(s_t, a_t))$$

After T steps, this induces a mapping from s_0 to s_T. The optimization problem is therefore:

$$\max_\phi \mathbb{E}_{s_0 \sim \mathcal{N}(0,I), \pi_\phi} \left[\sum_{t=0}^{T-1} \gamma^t r(s_t, a_t) \right],$$

with $r(s_t, a_t) = M\varepsilon(D(s_{t+1})) + C$, where s_0 is the initial latent and s_T is the latent after the policy rollout. The state-update function varies depending on the type of action space and is discussed in Section 3.3.

3.3 Action Space Variants

The action space of the RL problem is the core component of the proposed method. In simple terms, for each channel, the agent builds a direction vector and chooses an angle of rotation. Out of these c direction vectors and c angles, the algorithm builds a rotation matrix with c planes of rotation following Equation 2. The rotation matrix is then used to update each channel individually; if the latent space has only one channel, the rotation matrix reduces to a simple rotation. Furthermore, motivated by Hypothesis 3, this section presents two variants of RLSO: one with a continuous action space and the other with a discrete one. In the former, the RL agent outputs continuous-valued direction vectors and angles that define the rotation. In the discrete case, the agent outputs pairs of discrete values: in each pair, the first selects a direction vector from a predefined set of N_d direction vectors, and the second specifies the quantization bin for the angle value. Generally, the only variables between variants are the action space and, based on the dataset, the specific reward. All other architecture and implementation details are listed in Appendix E.2. The action spaces are formally defined in the following paragraphs.

Continuous Variant The actions lie in the space $a_t = (t_1, \ldots, t_c, \theta_1, \ldots, \theta_c)$, where t_i is a unit vector in the tangent space of the sphere at $s_{t,i}$ and $\theta_i \in [0, \pi]$ is a rotation angle. The next state is then computed for all channels:

$$s_{t+1} = f(s_t, a_t), \qquad f(s_t, a_t) = \begin{bmatrix} R^{(c)} s_{t,1} & \cdots & R^{(c)} s_{t,c} \end{bmatrix}$$

where $R^{(c)}$ is computed according to Equation 2 with $P_i = \begin{bmatrix} \hat{s}_{t,i}^\perp & t_i^\perp \end{bmatrix}$, where $s_{t,i}$ is channel i of s_t.

Discrete Variant The actions lie in the space $a_t = (j_1, \ldots, j_c, p_1, \ldots, p_c)$, where $j_i \in \{1, \ldots, N_d\}$ and $\theta_i = \frac{p_i \pi}{79}$, for $p_i \in 0, \ldots, 79$. The next state is then computed according to Equation 4:

$$s_{t+1} = f(s_t, a_t), \qquad f(s_t, a_t) = \begin{bmatrix} R^{(c)} s_{t,1} & \cdots & R^{(c)} s_{t,c} \end{bmatrix}$$

where $R^{(c)}$ is computed according to Equation 2 with $P_i = \begin{bmatrix} \hat{s}_{t,i}^\perp & h_{j_i}^\perp \end{bmatrix}$, where h_{j_i} is column j_i of the direction matrix $H \in \mathbb{R}^{n \times N_d}$, θ_i is the discretized angle, and $s_{t,i}$ is the channel i of s_t.

H is constructed from normalized partial columns of a Hadamard matrix of the smallest order $m \geq \max(n, N_d)$ for which one exists, which are then projected onto the tangent space of the sphere at $s_{t,i}$. This setup has two reasons: firstly, Hadamard columns are maximally distant from all coordinate hyperplanes and mutually orthogonal [53]. This is important because we are interested in transformations that affect a large portion of a latent vector's dimensions, since quality metrics are generally measured on the entire sample. However, simple rotations in arbitrary planes do not consistently achieve this. For example, Givens rotations [54] only modify the corresponding coordinate pair. They therefore induce negligible change in high-dimensional settings, causing the LDM to denoise the transformed latent sample into one nearly identical to the original, except for a small localized change. At the opposite extreme, one can show that rotations defined by planes spanned by vectors maximally distant from the coordinate hyperplanes impact the greatest number of dimensions of an n-dimensional vector. Secondly, Hadamard columns are evenly distributed across the search space, enabling uniform exploration.

4 Experimental Setup

To demonstrate the versatility of the proposed method, the experimental setup spans two model architectures and three different application fields. The first is an unconditional LDM trained on two different datasets; the second is an open-source, open-weight text-conditioned LDM. This section first describes the two model architectures with their corresponding datasets, then the two benchmarks that, in addition to the vanilla LDM, form the evaluation baselines. This setup, together with the ablation study in Appendix A, aims to validate three hypotheses:

Hypothesis 1. *Reinforcement learning is a viable paradigm for performing amortized, gradient-free optimization in the latent space of frozen diffusion models.*

Hypothesis 2. *Algorithmic exploration techniques that account for the inherent spherical geometry of Gaussian latent spaces significantly outperform naive approaches.*

Hypothesis 3. *Exploration techniques that enable control over the trade-off between generalization capabilities and computational efficiency offer significant advantages.*

4.1 Unconditional Latent Diffusion Model

The implementation of the unconditional LDM was adapted from von Platen et al. [55] and modified to include a variational autoencoder (VAE) with KL loss [56]. To ensure a unique data sample for each latent vector, we use the denoising diffusion implicit models (DDIM) [57] sampler. Architecture and training details are listed in Appendix E. The model is trained from scratch on two datasets from different domains: solutions of the Darcy Flow PDEs and an ad hoc synthetic image dataset. Both provide computable metrics that measure how much a sample violates the dataset validity constraints, i.e., a *residual error*. A residual error of zero corresponds to a valid sample. The residual errors are then negated to serve as a reward function, i.e., $r(s, a) = -\varepsilon(x)$.

Darcy Flow Here, the training data is a dataset created by Bastek et al. [58]: it contains 10000 solutions of the steady-state 2D Darcy-flow PDEs, which describe fluid movement through a porous medium. Each of the samples is generated by sampling the permeability field $K(\xi)$ from a Gaussian random field on a 64×64 grid and solving for the pressure distribution $p(\xi)$ with a finite-difference least-squares linear solver. This results in samples $(K, p) \in \mathbb{R}^{2\times 64\times 64}$. The per-grid-cell residual error is calculated based on the physical law of mass conservation as follows:

$$R(K, p) = \nabla \cdot (K \nabla p) + f. \tag{5}$$

where K is the permeability field, p is the pressure field, and f is the source function. A scalar residual error for one sample x_0 is then obtained as the mean absolute residual error:

$$\varepsilon(x) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} |R_{ij}(K, p)| \tag{6}$$

where $n = 64$ and $R_{ij}(K, p)$ is the residual error at grid cell (i, j). Further details can be found in Bastek et al. [58]. The reward threshold is set at $\tau = -0.4$.

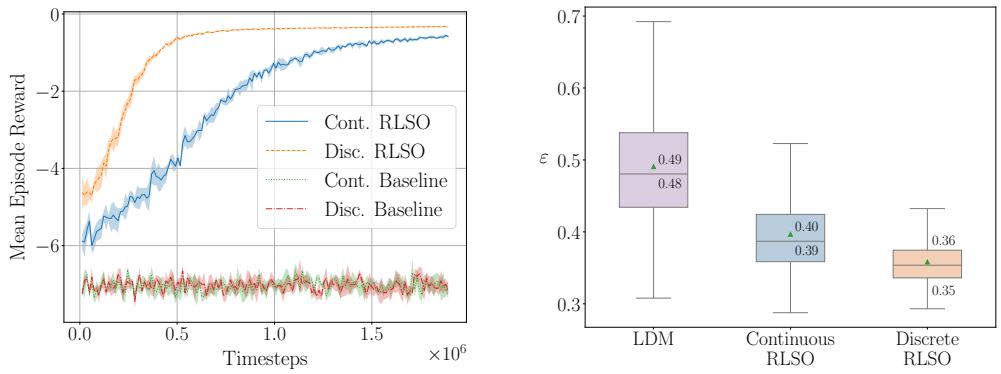


Figure 1: Darcy Flow: RL training over 5 seeds (left) and residual error comparison evaluated on 10000 samples for one seed (right).

Voronoi Deshpande et al. [59] introduce several synthetic datasets to benchmark image-synthesis generative models. These datasets, called Stochastic Context Models (SCM), contain images with different features, constraints, and rules that can be recovered after generation. They also provide scripts to compute a variety of quality metrics based on these features. To match the spatial dimensions of the Darcy Flow samples, we select a slightly simplified version of the Voronoi SCM for our experiments, using 64x64 instead of 256x256 images and restricting it to the class of images containing 16 regions. This creates a dataset of 10000 grayscale images. The reward is constructed out of a selection of the quality metrics introduced by the authors:

$$\varepsilon(x) = \frac{\mu_1}{A} + \frac{\sigma_1}{2B} + \frac{\mu_2}{\Gamma} + \frac{\sigma_2}{2\Delta} + \rho_{\mathrm{K}} + \rho_{\mathrm{S}} + 1.5\eta, \tag{7}$$

μ_1 and σ_1 are measures of the straightness of region edges, μ_2 and σ_2 are measures of the intra-region grayscale variance, ρ_{K} and ρ_{S} are Kendall's and Spearman's correlation errors between the regions' and the target grayscale values, and η is the error in region count. Furthermore, A, B, Γ, and Δ are normalizing constants. Specifically, $A = 0.0962016$ and $B = 0.116852$ are the average μ_1 and σ_1 of the training dataset, respectively. $\Gamma = 50$ and $\Delta = 20$ are set such that $\sim$95% of the non-zero values of the training dataset fall in the $[0, 1]$ range. The reward threshold was set at $\tau = -1.5$.

4.2 Text-Conditioned Latent Diffusion Model

The text-conditioned LDM experiments use Stable Diffusion 1.5 [2], integrated with the 2-Step Hyper-SD LoRA [60] and the DDIM sampler [57]. The chosen metric for sample quality is ImageReward [61], a pre-trained text-to-image human preference reward model that, given a generated image and its corresponding prompt, outputs a human preference score. Unlike the experiments on the unconditional LDM, the ImageReward score has no theoretical upper bound, and a higher score corresponds to a higher sample quality. To turn the ImageReward score into a reward function, the score is shifted so that most of $r(x)$ is negative. In this case, $r(x) = \mathrm{ImageReward}(D(x)) - 2$.

4.3 Baselines

To isolate the effect of the rotation-based exploration, the methods are evaluated with two baselines: one for the variant with a continuous action space and one for the variant with a discrete one.

Continuous Baseline This baseline is developed to offer a counterpart to the constrained exploration of the rotation-based methods, but still match their decision-making process. That is, the agent moves by choosing a unit direction vector and a scalar magnitude separately, which then get combined into a single perturbation vector. Specifically, the actions lie in the space $(u_1, \ldots, u_c, \alpha_1, \ldots, \alpha_c)$, where $\alpha_i \in \mathbb{R}$ is a scalar, $u_i \in \mathbb{R}^n$ is a unit vector, and $w_i = \alpha_i u_i \in \mathbb{R}^n$. The next state is then computed as: $s_{t+1} = [w_1 + s_{t,1} \quad \cdots \quad w_c + s_{t,c}]$.

Discrete Baseline The baseline for the discrete case is analogous: the agent chooses direction vectors u_i out of a set of N_d fixed ones and selects step-sizes α_i. The resulting vectors $w_i = \alpha_i u_i$ are then added to the corresponding channels. The step-size range is scaled

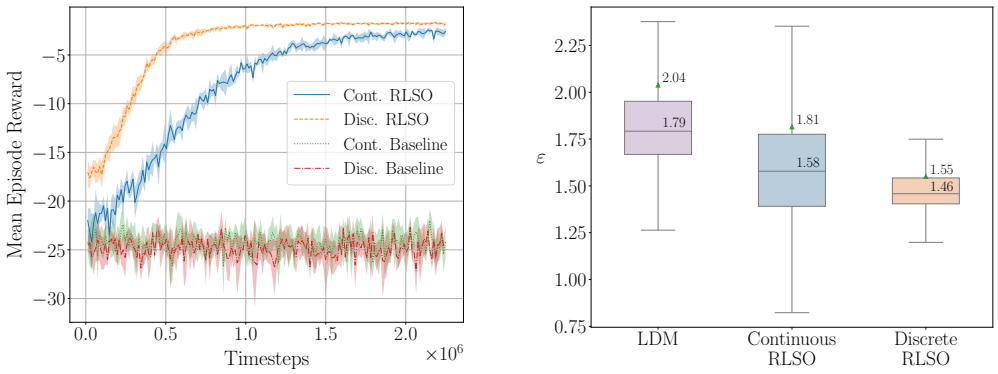


Figure 2: Voronoi: RL training over 5 seeds (left) and residual error comparison evaluated on 10000 samples for one seed (right).

so that the maximum attainable L_2-distance approximately matches that of the rotation-based methods, i.e., $2\sqrt{n}$. Specifically, the actions lie in the space $(j_1, \ldots, j_c, \alpha_1, \ldots, \alpha_c)$, where $j_i \in \{1, \ldots, N_d\}$ and $\alpha_i = 2\sqrt{n}\frac{p_i}{79}$, for $p \in \{0, \ldots, 79\}$. The next state is then computed as: $s_{t+1} = [w_1 + s_{t,1} \quad \cdots \quad w_c + s_{t,c}]$.

5 Results and Discussion

5.1 Unconditional LDM

Darcy Flow Figure 1 presents the results of the methods for the LDM trained on the Darcy Flow dataset. The left subfigure highlights two key findings: Firstly, RLSO's exploration technique consistently outperforms unconstrained exploration, thereby validating Hypothesis 2. Secondly, restricting the search directions to a predefined set led to faster and higher convergence, thereby validating Hypothesis 3. The right subfigure provides evidence for Hypothesis 1: the proposed framework achieved a $\sim$27% relative reduction in PDE residual compared to the vanilla LDM and obtained an absolute PDE residual comparable to state-of-the-art models, outperforming CoCoGen [62] and PG-diffusion [63] but trailing PIDM [58], as reported by Bastek et al. [58].

Voronoi Figure 2 presents the results of the methods for the LDM trained on the Voronoi dataset. The findings closely parallel those from the Darcy Flow experiments: geometry-aware exploration consistently outperforms unconstrained exploration, resulting in an approximate $\sim$11% relative reduction in residual error for the continuous variant and an approximate $\sim$24% relative reduction for the discrete action variant (up to $\sim$41%, see Figure 7 in the appendix).

Diversity The quality–diversity tradeoff induced by aligning generative models to measurable quality metrics is well studied [64]: stronger alignment can reduce sample diversity, which can be a desirable effect when it removes outputs that don't meet the required quality criteria. To analyze this phenomenon in our method, we have conducted an extensive study (see Appendix D.1). The results show that diversity mainly depends on action and latent space size. Continuous action spaces preserved diversity in all experiments, while in the discrete variant, the tradeoff was governed by the size of the latent space and the width of the direction matrix H.

Amortization To evaluate the benefits of amortizing the latent space search with a trained policy, we additionally compare RLSO against a naive best-of-N search. To match RLSO, best-of-N requires $N = 80$ samples on Voronoi and $N = 300$ samples on Darcy Flow, substantially increasing the inference cost per optimized sample. Additional details are provided in Appendix A.4.

5.2 Text-Conditioned LDM

Here, the latent space is significantly larger (Stable Diffusion 1.5 has a latent space $\mathcal{X} \in \mathbb{R}^{64\times64\times4}$, i.e., 16384 dimensions). To prevent the action space from exploding in dimensionality, we use only

the discrete variant with a limited number of fixed directions ($N_d = 256$). The model is trained with the fixed prompt: *"A photo of an astronaut riding a horse on mars"*. Table 1 presents the results for the text-conditioned LDM. Similar to the unconditional LDM experiments, RLSO significantly improves the sample quality based on the target metric: in this case, the human preference score from the ImageReward model. When evaluated with the same prompt, RLSO achieves a mean absolute gain of 0.35. Further experiments, presented in Figure 8 in the appendix, show that gains can reach up to 1.26 when trained and evaluated on the same prompt. When evaluated on a different prompt and with random prompts from a small prompt dataset, the mean absolute gains are 0.12 and 0.01, respectively. Moreover, the model can cope with dropping the Hyper-SD LoRA with which it was trained at inference time, albeit with a slightly reduced effectiveness. To rule out reward hacking, we additionally evaluate RLSO using quality metrics independent of the ImageReward score: across 1000 samples for each of three prompts, RLSO improves CLIP score [65] in all cases, while CLIP-IQA [66] and NIQE [67] show only small, mixed changes, providing no evidence of quality degradation. Additionally, we compare RLSO against benchmarks from related work and find that it achieves a competitive reward-diversity tradeoff across two evaluated prompts. The full independent quality metric evaluation, the comparison with related-work benchmarks, and qualitative results are available in Appendices D.2, C, and F, respectively.

In addition to demonstrating the effectiveness of RLSO, this experiment demonstrates the importance of its modularity. Despite the significant increase in the dimensionality of the latent space, the number of training steps did not significantly increase. This is due to two reasons: firstly, training time scales primarily with the action space, which we can control independently of dimensionality by adjusting the number of directions; secondly, complexity scales only minimally with the number of channels due to the use of c-fold rotations.

Table 1: Text-conditioned LDM: ImageReward comparison evaluated on 10000 samples. Full distributions are shown in Figure 8(a) in the appendix.

Model	ImageReward (mean) ↑	ImageReward (median) ↑
LDM	1.28	1.56
RLSO	$\mathbf{1.63}$	$\mathbf{1.78}$

6 Limitations

RLSO is inherently shaped by the expressive capacity of the underlying diffusion model. It is particularly effective at identifying optimal latent codes within the learned data manifold, but it cannot introduce features outside the model's training distribution. Furthermore, the method assumes a new policy is trained for each new downstream task, which is most practical when the training cost can be amortized over sufficiently many generated samples. While our text-to-image results show some generalization, this is currently evaluated only in a limited setting. Finally, while our current implementation uses PPO, future work could explore more sample-efficient RL paradigms to further improve scalability.

7 Conclusion

In this work, we introduced Rotational Latent Space Optimization, a novel framework that leverages RL for efficient, geometry-aware guidance of pre-trained diffusion models. Our experiments, conducted across three diverse scientific domains, demonstrate that respecting the spherical geometry of the latent space via rotational exploration produces significantly higher-quality samples compared to unconstrained optimization. Moreover, by amortizing the optimization process with an RL agent, our method generates constraint-aligned samples in a single denoising pass, eliminating the substantial computational overhead of iterative LSO techniques. We have shown that this approach is a viable and effective paradigm for improving sample quality wherever a guiding reward signal is available, establishing a new path for efficient latent space control. Furthermore, we have shown that the modularity of our approach facilitates optimization in lower-dimensional subspaces, ensuring the method scales effectively even as latent space dimensionality and model complexity increase.

Acknowledgments and Disclosure of Funding

We acknowledge the financial support by the Bavarian State Ministry for Economic Affairs, Regional Development and Energy (StMWi) for the Lighthouse Initiative KI.FABRIK, (Phase 1: Infrastructure as well as the research and development program under grant no. DIK0249). The authors declare no competing interests.

References

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

[2] 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. doi: 10.1109/CVPR52688.2022.01042.

[3] Jingyu Hu, Ka-Hei Hui, Zhengzhe Liu, Ruihui Li, and Chi-Wing Fu. Neural wavelet-domain diffusion for 3d shape generation, inversion, and manipulation. *ACM Transactions on Graphics*, 43(2):1–18, 2024. doi: 10.1145/3635304.

[4] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621, 2019. doi: 10.1109/ICASSP.2019.8683143.

[5] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6199–6203, 2020. doi: 10.1109/ICASSP40776.2020.9053795.

[6] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021.

[7] Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018. doi: 10.1021/acscentsci.7b00572.

[8] Nicola De Cao and Thomas Kipf. Molgan: An implicit generative model for small molecular graphs. *arXiv preprint*, 2018. doi: 10.48550/arXiv.1805.11973.

[9] Chengxi Zang and Fei Wang. Moflow: an invertible flow model for generating molecular graphs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 617–626, 2020. doi: 10.1145/3394486.3403104.

[10] Ruoxi Sun, Hanjun Dai, Li Li, Steven Kearnes, and Bo Dai. Towards understanding retrosynthesis by energy-based models. *Advances in Neural Information Processing Systems*, 34: 10186–10194, 2021.

[11] Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. In *International Conference on Learning Representations*, 2022.

[12] Donatas Repecka, Vykintas Jauniskis, Laurynas Karpus, Elzbieta Rembeza, Irmantas Rokaitis, Jan Zrimec, Simona Poviloniene, Audrius Laurynenas, Sandra Viknander, Wissam Abuajwa, et al. Expanding functional protein sequence spaces using generative adversarial networks. *Nature Machine Intelligence*, 3(4):324–333, 2021. doi: 10.1038/s42256-021-00310-5.

[13] Elizaveta Kozlova, Arthur Valentin, Aous Khadhraoui, and Daniel Nakhaee-Zadeh Gutierrez. Proteinflow: a python library to pre-process protein structure data for deep learning applications. *bioRxiv*, 2023. doi: 10.1101/2023.09.25.559346.

[14] Joseph L. Watson, David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, Andrew J. Borst, Robert J. Ragotte, Lukas F. Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023. doi: 10.1038/s41586-023-06415-8.

[15] Yifeng Jiang, Tingnan Zhang, Daniel Ho, Yunfei Bai, C. Karen Liu, Sergey Levine, and Jie Tan. Simgan: Hybrid simulator identification for domain adaptation via adversarial reinforcement learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2884–2890, 2021. doi: 10.1109/ICRA48506.2021.9561731.

[16] Jungdam Won, Deepak Gopinath, and Jessica Hodgins. Physics-based character controllers using conditional vaes. *ACM Transactions on Graphics*, 41(4):1–12, 2022. doi: 10.1145/3528223.3530067.

[17] Benjamin Holzschuh, Simona Vegetti, and Nils Thuerey. Solving inverse physics problems with score matching. *Advances in Neural Information Processing Systems*, 36:61888–61922, 2023.

[18] Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Zilong Wang, Aliaksandra Shysheya, Jonathan Crabbé, Shoko Ueda, et al. A generative model for inorganic materials design. *Nature*, pages 624–632, 2025. doi: 10.1038/s41586-025-08628-5.

[19] Olinde Rodrigues. Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire. *Journal de mathématiques pures et appliquées*, 5:380–440, 1840.

[20] Arthur Cayley. *On the Motion of Rotation of a Solid Body*, page 28–35. Cambridge Library Collection - Mathematics. Cambridge University Press, 2009. doi: 10.1017/CBO9780511703676.007.

[21] Arthur Cayley. *Sur quelques propriétés des déterminants gauches.* de Gruyter, 1846.

[22] P. H. Schoute. *Le déplacement le plus général dans l'espace à n dimensions*. Brill, 1892.

[23] Leonhard Euler. Formulae generales pro translatione quacunque corporum rigidorum. *Novi Commentarii Academiae Scientiarum Petropolitanae*, pages 189–207, 1776.

[24] Daniele Mortari. On the rigid rotation concept in n-dimensional spaces. *The Journal of the Astronautical Sciences*, 49:401–420, 2001. doi: 10.1007/BF03546230.

[25] Daniele Mortari. n-dimensional cross product and its application to the matrix eigenanalysis. *Journal of Guidance, Control, and Dynamics*, 20(3):509–515, 1997. doi: 10.2514/3.60598.

[26] Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge University Press, 2019. doi: 10.1017/9781108627771.

[27] Avrim Blum, John Hopcroft, and Ravindran Kannan. *Foundations of data science*. Cambridge University Press, 2020.

[28] Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: on the curvature of deep generative models. In *International Conference on Learning Representations*, 2018.

[29] Nutan Chen, Alexej Klushyn, Richard Kurle, Xueyan Jiang, Justin Bayer, and Patrick Smagt. Metrics for deep generative models. In *International Conference on Artificial Intelligence and Statistics*, pages 1540–1550, 2018.

[30] Mathurin Videau, Nickolai Knizev, Alessandro Leite, Marc Schoenauer, and Olivier Teytaud. Interactive latent diffusion model. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 586–596, 2023. doi: 10.1145/3583131.3590471.

[31] Dvir Samuel, Rami Ben-Ari, Nir Darshan, Haggai Maron, and Gal Chechik. Norm-guided latent space exploration for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:57863–57875, 2023.

[32] Erik Bodin, Alexandru I. Stere, Dragos D. Margineantu, Carl Henrik Ek, and Henry Moss. Linear combinations of latents in generative models: subspaces and beyond. In *International Conference on Learning Representations*, 2025.

[33] Luca Sacchetto, Stefan Röhrl, and Klaus Diepold. Iterative visual interaction with latent diffusion models. In *International Conference on Human-Computer Interaction*, pages 411–421, 2024. doi: 10.1007/978-3-031-60606-9_24.

[34] Yong-Hyun Park, Mingi Kwon, Junghyo Jo, and Youngjung Uh. Unsupervised discovery of semantic latent directions in diffusion models. *arXiv preprint*, 2023. doi: 10.48550/arXiv.2302.12469.

[35] Cheng Jin, Zhenyu Xiao, Chutao Liu, and Yuantao Gu. Angle domain guidance: Latent diffusion requires rotation rather than extrapolation. In *Proceedings of the 42nd International Conference on Machine Learning*, pages 28187–28212, 2025.

[36] Shyamgopal Karthik, Karsten Roth, Massimiliano Mancini, and Zeynep Akata. If at first you don't succeed, try, try again: Faithful diffusion-based text-to-image generation by selection. *arXiv preprint*, 2023. doi: 10.48550/arXiv.2305.13308.

[37] Bram Wallace, Akash Gokul, Stefano Ermon, and Nikhil Naik. End-to-end diffusion latent optimization improves classifier guidance. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7246–7256, 2023. doi: 10.1109/ICCV51070.2023.00669.

[38] Dvir Samuel, Rami Ben-Ari, Simon Raviv, Nir Darshan, and Gal Chechik. Generating images of rare concepts using pre-trained diffusion models. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, 2024. doi: 10.1609/aaai.v38i5.28270.

[39] Korrawe Karunratanakul, Konpat Preechakul, Emre Aksan, Thabo Beeler, Supasorn Suwajanakorn, and Siyu Tang. Optimizing diffusion noise can serve as universal motion priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1334–1345, 2024. doi: 10.1109/CVPR52733.2024.00133.

[40] Luca Eyring, Shyamgopal Karthik, Karsten Roth, Alexey Dosovitskiy, and Zeynep Akata. Reno: Enhancing one-step text-to-image models through reward-based noise optimization. *Advances in Neural Information Processing Systems*, 37:125487–125519, 2024.

[41] Qiucheng Wu, Yujian Liu, Handong Zhao, Trung Bui, Zhe Lin, Yang Zhang, and Shiyu Chang. Harnessing the spatial-temporal attention of diffusion models for high-fidelity text-to-image synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7766–7776, 2023. doi: 10.1109/ICCV51070.2023.00714.

[42] Cheng Lu, Huayu Chen, Jianfei Chen, Hang Su, Chongxuan Li, and Jun Zhu. Contrastive energy prediction for exact energy-guided diffusion sampling in offline reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, pages 22825–22855, 2023.

[43] Donghoon Ahn, Jiwon Kang, Sanghyun Lee, Jaewon Min, Minjae Kim, Wooseok Jang, Hyoungwon Cho, Sayak Paul, SeonHwa Kim, Eunju Cha, Kyong Hwan Jin, and Seungryong Kim. A noise is worth diffusion guidance. In *International Conference on Learning Representations*, 2026.

[44] Siddarth Venkatraman, Mohsin Hasan, Minsu Kim, Luca Scimeca, Marcin Sendera, Yoshua Bengio, Glen Berseth, and Nikolay Malkin. Outsourced diffusion sampling: Efficient posterior inference in latent spaces of generative models. In *Proceedings of the 42nd International Conference on Machine Learning*, pages 61212–61239, 2025.

[45] Luca Eyring, Shyamgopal Karthik, Alexey Dosovitskiy, Nataniel Ruiz, and Zeynep Akata. Noise hypernetworks: Amortizing test-time compute in diffusion models. *Advances in Neural Information Processing Systems*, 38:95429–95462, 2025.

[46] Andrew Wagenmaker, Yunchu Zhang, Mitsuhiko Nakamoto, Seohong Park, Waleed Yagoub, Anusha Nagabandi, Abhishek Gupta, and Sergey Levine. Steering your diffusion policy with latent space reinforcement learning. In *9th Annual Conference on Robot Learning*, 2025.

[47] Jing Hu, Chengming Feng, Shu Hu, Ming-Ching Chang, Xin Li, Xi Wu, and Xin Wang. Rlministyler: Light-weight rl style agent for arbitrary sequential neural style generation. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 1116–1124, 2025. doi: 10.24963/ijcai.2025/125.

[48] Shiyuan Zhang, Weitong Zhang, and Quanquan Gu. Energy-weighted flow matching for offline reinforcement learning. In *International Conference on Learning Representations*, 2025.

[49] Pertti Lounesto. *Clifford Algebras and Spinors*. Cambridge University Press, 2 edition, 2001. doi: 10.1017/CBO9780511526022.

[50] Erhard Schmidt. Über die auflösung linearer gleichungen mit unendlich vielen unbekannten. *Rendiconti del Circolo Matematico di Palermo (1884-1940)*, 25(1):53–77, 1908. doi: 10.1007/978-3-322-84410-1_5.

[51] Persi Diaconis and David Freedman. Asymptotics of graphical projection pursuit. *The Annals of Statistics*, pages 793–815, 1984. doi: 10.1214/aos/1176346703.

[52] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint*, 2017. doi: 10.48550/arXiv.1707.06347.

[53] Wojciech Tadej and Karol Życzkowski. A concise guide to complex hadamard matrices. *Open Systems & Information Dynamics*, 13(2):133–177, 2006. doi: 10.1007/s11080-006-8220-2.

[54] Wallace Givens. Computation of plain unitary rotations transforming a general matrix to triangular form. *Journal of the Society for Industrial and Applied Mathematics*, 6(1):26–50, 1958. doi: 10.1137/0106004.

[55] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. `github.com/huggingface/diffusers`, 2022.

[56] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

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

[58] Jan-Hendrik Bastek, WaiChing Sun, and Dennis Kochmann. Physics-informed diffusion models. In *International Conference on Learning Representations*, 2025.

[59] Rucha Deshpande, Mark A. Anastasio, and Frank J. Brooks. A method for evaluating deep generative models of images for hallucinations in high-order spatial context. *Pattern Recognition Letters*, 186:23–29, 2024. doi: 10.1016/j.patrec.2024.08.023.

[60] Yuxi Ren, Xin Xia, Yanzuo Lu, Jiacheng Zhang, Jie Wu, Pan Xie, Xing Wang, and Xuefeng Xiao. Hyper-sd: Trajectory segmented consistency model for efficient image synthesis. *Advances in Neural Information Processing Systems*, 37:117340–117362, 2024.

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

[62] Christian Jacobsen, Yilin Zhuang, and Karthik Duraisamy. Cocogen: Physically consistent and conditioned score-based generative models for forward and inverse problems. *SIAM Journal on Scientific Computing*, 47(2):C399–C425, 2025. doi: 10.1137/24M1636071.

[63] Dule Shu, Zijie Li, and Amir Barati Farimani. A physics-informed diffusion model for high-fidelity flow field reconstruction. *Journal of Computational Physics*, 478:111972, 2023. doi: 10.1016/j.jcp.2023.111972.

[64] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

[65] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763, 2021.

[66] Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, volume 37, pages 2555–2563, 2023. doi: 10.1609/aaai.v37i2.25353.

[67] Anish Mittal, Rajiv Soundararajan, and Alan C. Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212, 2013. doi: 10.1109/LSP.2012.2227726.

[68] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015. doi: 10.1007/978-3-319-24574-4_28.

[69] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. doi: 10.1109/ICCV51070.2023.00387.

[70] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[71] Stephen D. Liang. Reconsidering noise for denoising diffusion probabilistic models. In *NeurIPS 2025 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2025.

[72] Siddarth Venkatraman, Moksh Jain, Luca Scimeca, Minsu Kim, Marcin Sendera, Mohsin Hasan, Luke Rowe, Sarthak Mittal, Pablo Lemos, Emmanuel Bengio, et al. Amortizing intractable inference in diffusion models for vision, language, and control. *Advances in Neural Information Processing Systems*, 37:76080–76114, 2024.

[73] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, 2024.

[74] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36:79858–79885, 2023.

[75] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[76] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

A Ablation Study

A.1 Projection-based Exploration

To further validate the hypothesis that rotations constitute an efficient exploration strategy, we compare RLSO against a popular norm-preserving optimization method used in LSO (similar to, for example, Wallace et al. [37]). We construct a baseline with an action space analogous to that of the discrete variant; however, instead of applying a rotation, this baseline projects the update back onto the hypersphere. The actions lie in the space (j, α), where $j \in \{1, \ldots, N_d\}$ and $\alpha = \frac{p}{C}$, for $p \in \{0, \ldots, 79\}$. The next state is then computed according to:

$$s_{t+1} = \sqrt{n} \cdot \frac{s_t + h_j \alpha}{\|s_t + h_j \alpha\|_2},$$

where h_j is the j-th column of the direction matrix $H \in \mathbb{R}^{n \times N_d}$, C controls step size, and $n = 256$ denotes the dimensionality of the latent space. In RLSO, step-size has a direct geometric interpretation and is naturally bounded by $[0, \pi]$. For the projection-based ablation, the choice of step-size range is less straightforward, since α could potentially be defined over the whole of $\mathbb{R}$. We heuristically set $C = 4.9375$, such that $\max(\alpha) = \sqrt{n} = 16$ approximately matches the norm of the latent vectors. Figure 3 demonstrates that RLSO significantly outperforms this norm-constrained baseline. We attribute this performance gap to the limitations of projection-based exploration, which is inherently restricted in its angular reach compared to rotational updates.

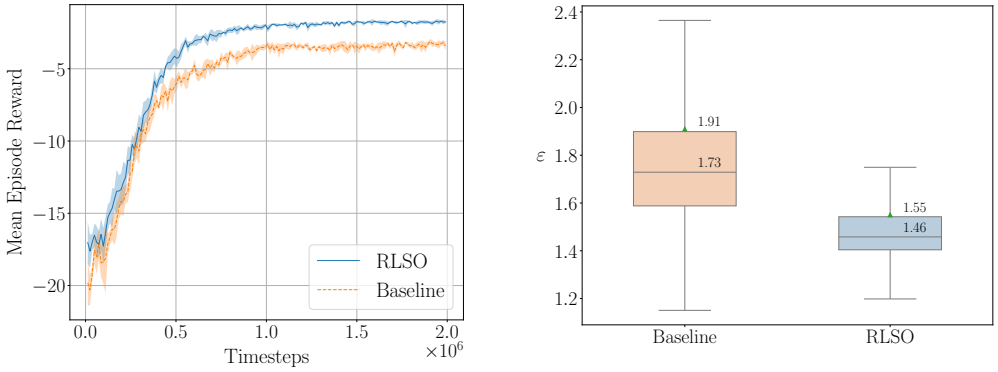


Figure 3: Voronoi: comparison between RLSO and projection-based baseline: training over 5 seeds (left) and residual error comparison evaluated on 10000 samples for one seed (right).

A.2 Isoclinic Rotations

The aim of this ablation is to evaluate whether using an isoclinic rotation instead of a c-fold rotation affects performance. As discussed in Section 3.1, isoclinic rotations are a special form of c-fold rotations, where the angles of all active planes of rotation are equal. To that end, we construct a baseline similar to the discrete action variant, except that the angle is shared across channels. The actions lie in the space

$$a_t = (j_1, \ldots, j_c, \theta),$$

where $j_i \in \{1, \ldots, N_d\}$ and $\theta = \frac{p\pi}{79}$, for $p \in \{0, \ldots, 79\}$ is the singular rotation angle. The next state is then computed according to Equation 4:

$$s_{t+1} = f(s_t, a_t)$$

with

$$f(s_t, a_t) = \begin{bmatrix} R^{(c)} s_{t,1} & \cdots & R^{(c)} s_{t,c} \end{bmatrix}$$

where $R^{(c)}$ is computed according to Equation 2 with:

$$P_i = \begin{bmatrix} \hat{s}_{t,i}^{\perp} & h_{j_i}^{\perp} \end{bmatrix},$$

where h_{j_i} denotes column j_i of the direction matrix $H \in \mathbb{R}^{n \times N_d}$, θ is the discretized angle, $s_{t,i}$ is the channel i of s_t.

In an experiment performed on the Darcy Flow dataset, the results show that the effect is minimal: the isoclinic rotation converges slightly faster and achieves a slightly higher average residual error ($\sim 1.7\%$). This may stem mostly from the reduced complexity due to the fewer free parameters.

A.3 One-Step Contextual Bandit

As mentioned in Section 3.2, a contextual bandit formulation would be mathematically sufficient, since the goal is to identify an optimal sample in one or a few transformations, not to find an optimal trajectory. However, we found the multi-step MDP formulation to be empirically superior. To evaluate this, we perform two experiments, one on Darcy Flow and one on Stable Diffusion, comparing RLSO against a controlled one-step PPO baseline. This baseline matches the multi-step setup in policy architecture, action parameterization, optimization, training budget, and evaluation protocol, changing only the episode horizon to $T = 1$.

Table 2: Comparison between RLSO and the one-step contextual bandit baseline on Darcy Flow

Model	Residual ↓	W_K ↓	JS_K ↓	W_p ↓	JS_p ↓
One-step contextual bandit	$\mathbf{0.30}$	0.151	0.024	0.006	0.02
RLSO (multi-step MDP)	0.36	$\mathbf{0.006}$	$\mathbf{0.005}$	$\mathbf{0.004}$	$\mathbf{0.009}$

W: Wasserstein distance between generated permeability samples (W_K) and pressure samples (W_p). JS: Jensen-Shannon divergence between generated permeability samples (JS_K) and pressure samples (JS_p).

Table 3: Comparison between RLSO and the one-step contextual bandit baseline on Stable Diffusion with the prompt *A photo of an astronaut riding a horse on mars.*

Model	ImageReward ↑	CLIP diversity ↑
One-step contextual bandit	1.378	$\mathbf{0.14}$
RLSO (multi-step MDP)	$\mathbf{1.63}$	0.12

The results reported in Tables 2 and 3 demonstrate the advantages of multi-step RL, while revealing two distinct regimes. In the smaller Darcy Flow latent space, the one-step formulation optimizes more aggressively but causes a severe shift in the output distribution. Multi-step PPO yields a substantially better quality-diversity trade-off. On Stable Diffusion, the one-step formulation converges to significantly lower rewards, suggesting that a single transformation is less effective at exploring the much larger latent space.

We hypothesize that these different behaviors arise from two phenomena: In the larger latent space of Stable Diffusion, the one-step formulation has access to a restricted class of transformations. Multiple steps provide a richer, more effective transformation class without requiring an impractically large one-step action space. Sequential updates also allow the policy to refine the latent over several smaller transformations, while intermediate evaluations provide denser training signals. In the smaller Darcy Flow latent space, the transformations available to one-step PPO appear expressive enough to reach higher rewards, but its setup doesn't include a mechanism to prevent distribution shift. In the multi-step setup, the reward threshold regularizes how aggressively the policy moves and stops optimization once sufficient quality is reached.

A.4 Best-of-N Search

To evaluate the benefit of amortizing the latent space search with a learned policy, we perform an experiment comparing RLSO with the unconditional LDM against a best-of-N baseline. To that end, we generate N candidates using the same frozen diffusion models that serve as inputs to RLSO and select the candidate with the highest score. We repeat this procedure to obtain 1000 output samples and report their mean residual. The results shown in Table 4 isolate the advantage of the amortization: to match RLSO on the Voronoi dataset, best-of-N requires 80 passes through the diffusion model; to match RLSO on the Darcy Flow dataset, best-of-N requires 300. Additionally, together with the training details listed in Appendix E.1, these values can be used to calculate the amortization

break-even points. These are given by the formula $N_{\text{train}}/(N-1)$, which results in break-even points of 2007 generated samples for Darcy Flow and 10127 for Voronoi.

Table 4: Comparison between the discrete variant of RLSO and the best-of-N baseline on Darcy Flow and Voronoi

Model	Voronoi Residual ↓	Darcy Flow Residual ↓
RLSO (discrete variant)	1.32	0.33
Best-of-10	1.48	0.40
Best-of-80	1.32	0.36
Best-of-300	1.24	0.33

A.5 Diffusion Backbone

While conventional diffusion models use a U-Net [68] as a denoising backbone, some newer models use a transformer-based denoising architecture instead. Since our method operates on the initial Gaussian noise and is motivated by its geometry, we assume it to be backbone-agnostic. To support this assumption, we perform an experiment on a pretrained class-conditional Diffusion Transformer (DiT) backbone [69], trained on ImageNet [70]. Specifically, we use the class *white shark* and train RLSO with ImageReward using the prompt *a picture of a white shark*. As shown in Table 5, RLSO improves the mean ImageReward score over the vanilla DiT baseline by ~ 1.1 (computed on 1000 samples).

Table 5: RLSO performance on DiT backbone

Model	ImageReward ↑
DiT	-0.028 ± 1.187
DiT+RLSO (ours)	$\mathbf{1.071 \pm 0.431}$

B Distributional Effect of RLSO

The state-dependent rotations induced by RLSO preserve the norm of transformed latent samples by construction, keeping the samples in the Gaussian shell. However, they need not preserve the full Gaussian law. To measure the distributional shift induced by the policy, we run the inference procedure of two trained instances of RLSO on 1000 initial noise samples $s_0 \sim \mathcal{N}(0, I)$: the continuous variant on Darcy Flow and the discrete variant on Stable Diffusion. We record the average empirical mean, skewness, excess kurtosis, and norm. Moreover, we run a per-sample Kolmogorov-Smirnov (KS) test.

Table 6: Distributional shift induced by RLSO

Statistic	Continuous RLSO ($n = 256$)	Discrete RLSO ($n = 4096$)
Norm	15.997	63.913
Empirical mean	0.012 ± 0.034	0.001 ± 0.005
Empirical skewness	0.002 ± 0.075	-0.0002 ± 0.106
Empirical excess kurtosis	-0.004 ± 0.163	-1.08 ± 0.13
KS statistic	0.055 ± 0.016	0.125 ± 0.105
KS rejection rate	0.058	0.774

The results are shown in Table 6. For the continuous variant, all statistics are consistent with a Gaussian: mean, skewness, and excess kurtosis are close to zero relative to their observed across-sample variability, and the fraction of vectors rejected under the per-vector KS test matches the rate expected under a true null.

For the discrete variant, mean and skewness remain consistent with zero, but excess kurtosis is significantly negative, and the KS test rejected 77.4% of samples. This is consistent with the approximate Gaussian mixture that a rotation of a Gaussian toward a Hadamard column generally produces. However, exact preservation of the Gaussian law is not necessarily required for diffusion sampling. Liang [71, Theorem 5] proves that, instead, diffusion sampling can remain well-behaved under non-Gaussian initial distributions satisfying broader conditions, including symmetry, zero mean, finite second moment, and the existence of a density. Under rotations, norm preservation is exact, giving a finite second moment. Indeed, we found that as long as the direction matrix H contains balanced, zero-mean columns, the rotation will also produce a balanced, zero-mean sample.

C Text-conditioned LDM: Literature Anchoring of ImageReward Scores

In their paper, Venkatraman et al. [72] introduce Relative Trajectory Balance (RTB) and include a comparison to previous works that use RL to train or fine-tune diffusion models, including DDPO [73] and DPOK [74]. In Figure 3 of Venkatraman et al. [72], the authors report the average ImageReward score and the average diversity score (measured as the mean cosine distance between CLIP embeddings) for each model across four fixed prompts. To compare RLSO with these benchmarks, we retrain it on Stable Diffusion 1.5, i.e., by dropping the Hyper-SD LoRA, and evaluate it on two of the prompts. Thus, the experiment matches the backbone, guidance scale, number of inference steps, and scheduler used by the authors.

Table 7: Performance of RLSO compared to benchmarks from Venkatraman et al. [72] on the prompt *"A green colored rabbit."*

Model	ImageReward ↑	CLIP diversity ↑
Prior	-0.113	0.144
DDPO	1.798	0.049
DPOK	1.492	0.076
RTB	1.756	0.044
Prior (ours)	-0.269	0.141
RLSO	0.769	0.119
RLSO (greedy)	1.780	0.074

Note: the values for the author's prior, DDPO, DPOK, and RTB were extracted from Figure 3 in Venkatraman et al. [72].

Table 8: Performance of RLSO compared to benchmarks from Venkatraman et al. [72] on the prompt *"Four roses."*

Model	ImageReward ↑	CLIP diversity ↑
Prior	-0.374	0.245
DDPO	1.095	0.120
DPOK	1.467	0.190
RTB	1.423	0.157
Prior (ours)	-0.377	0.217
RLSO	1.031	0.188
RLSO (greedy)	1.448	0.066

Note: the values for the author's prior, DDPO, DPOK, and RTB were extracted from Figure 3 in Venkatraman et al. [72].

We additionally evaluate a greedy RLSO variant, in which the highest-probability action is selected instead of sampling from the policy. This requires no retraining and provides a higher-reward operating point for comparison with the more strongly optimized outputs of the other methods (all other evaluations in this work use sampled actions, which preserve greater sample diversity). As shown in Tables 7 and 8, RLSO achieves a competitive reward–diversity trade-off across the two prompts.

D Additional Analysis of Quality and Diversity

D.1 Tradeoff between Quality and Diversity

To investigate the trade-off between quality and diversity, this section analyzes the effects of several variables on the diversity and quality of the generated samples from the unconditional LDM. Specifically, the number of directions N_d (continuous, 256, or 512); the angle range ($(0, \pi)$, $[0, \pi]$ or $[0, \frac{\pi}{2}]$); and the number of discretization bins B_θ (80 or 160).

D.1.1 Darcy Flow

We evaluate diversity by generating 1000 Darcy Flow samples for each experiment, aggregating them by channel (permeability K and pressure p) and computing a kernel-density estimation. We evaluate sample quality by generating 10000 samples and evaluating the residual error ε. Figure 4 shows the results of the diversity evaluation, and Figure 5 shows the results of the quality evaluation. These highlight several key findings:

1. RLSO with a continuous action space (see Section 3.3) achieves an increase in sample quality without significant loss of diversity.
2. Discretizing and limiting the number of directions (i.e., the discrete variant described in Section 3.3) leads to improved sample quality, but can lead to significant loss in diversity beyond a certain threshold.
3. Angle range and number of discretization bins have little effect on quality or diversity.

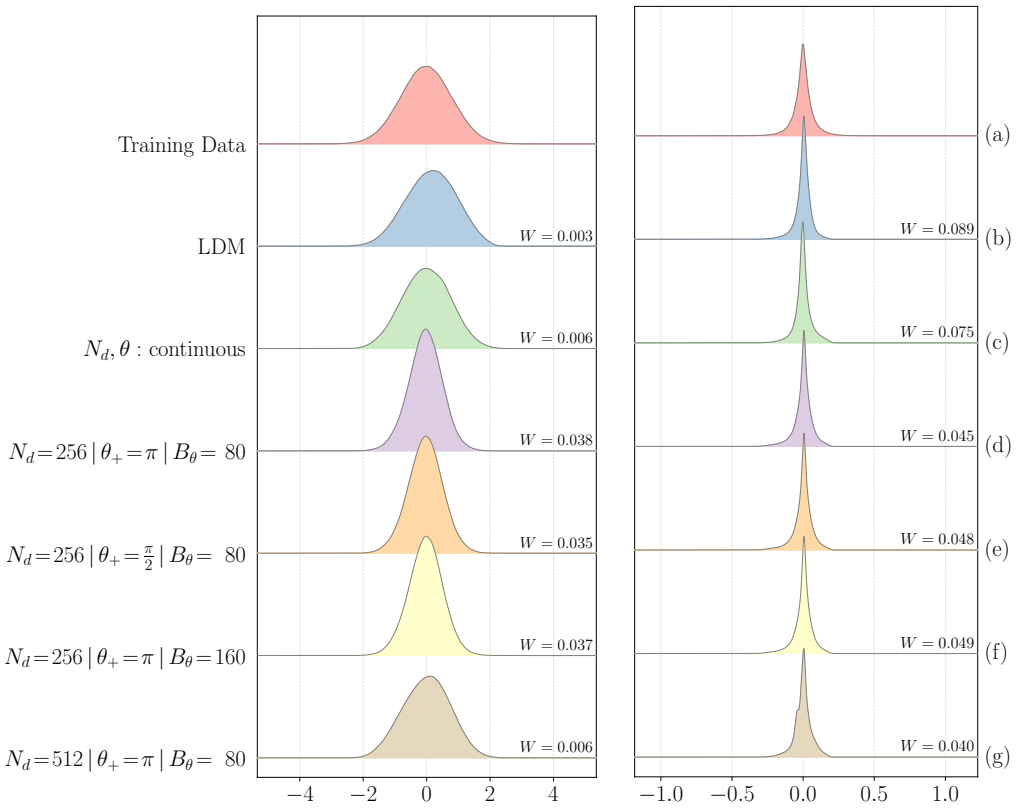


Figure 4: Darcy Flow: Distribution of permeability K (left) and pressure p (right) for different numbers of directions N_d, angle ranges $[0, \theta_+]$, and number of angle discretization bins B_θ. W denotes the Wasserstein distance to the training data. Plot (c) corresponds to the continuous variant; plots (d,e,f,g) correspond to different configurations of the discrete variant.

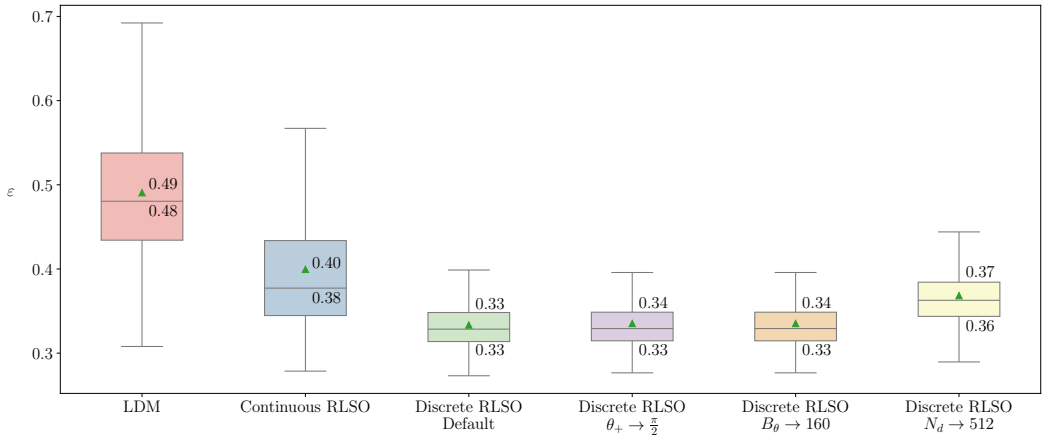


Figure 5: Darcy Flow: residual error for different numbers of directions N_d, angle ranges $[0, \theta_+]$, and number of angle discretization bins B_θ.

D.1.2 Voronoi

Here, we evaluate diversity by generating 1000 images, automatically detecting the position of the region centroids and plotting their distribution on the 64×64 pixel grid. Similarly to the Darcy Flow dataset, we evaluate sample quality by generating 10000 samples and evaluating the residual error ε.

Figure 6 shows the results of the diversity evaluation; Figure 7 shows the results of the quality evaluation. These partly mimic the results from Section D.1.1, with a few important differences:

1. Discretizing direction and angle always leads to significant loss in diversity.
2. Angle range and number of discretization bins have a meaningful effect on both sample quality and diversity.

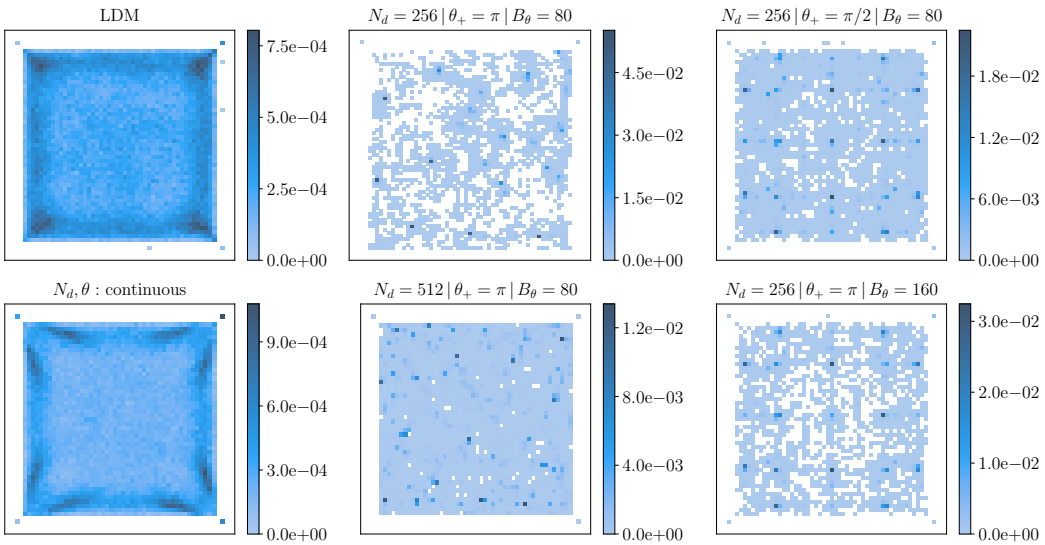


Figure 6: Voronoi: Distribution of region centroids for different numbers of directions N_d, angle ranges $[0, \theta_+]$, and number of angle discretization bins B_θ. The default configuration of the discrete RLSO (upper row, central image) is: $\theta_+ = \pi$, $N_d = 256$, $B_\theta = 80$.

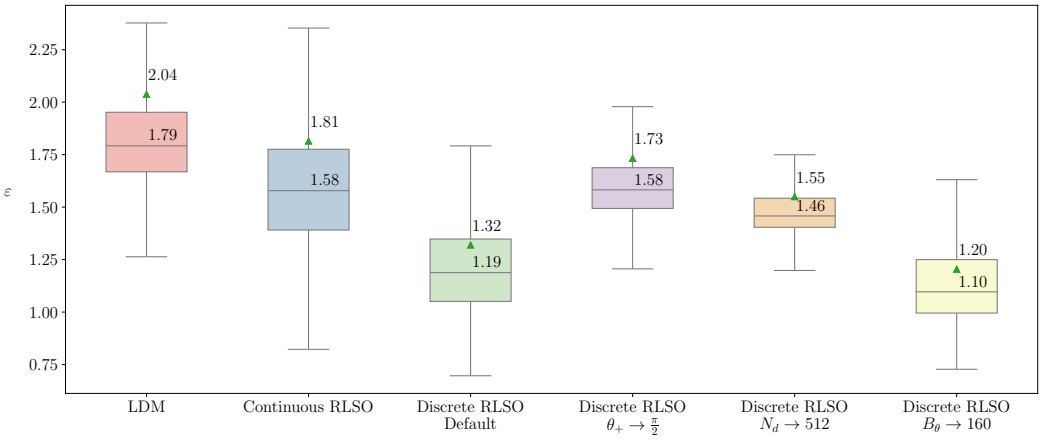


Figure 7: Voronoi: residual error for different numbers of directions N_d, angle ranges $[0, \theta_+]$, and number of angle discretization bins B_θ. The default configuration of the discrete RLSO is: $\theta_+ = \pi$, $N_d = 256$, $B_\theta = 80$.

D.2 Text-conditioned LDM

To evaluate quality independently of the model's reward and rule out potential reward hacking, we compare RLSO with the vanilla LDM using three additional metrics: CLIP score [65], which measures prompt faithfulness, and two no-reference image quality metrics, CLIP Image Quality Assessment (CLIP-IQA) [66] and the Naturalness Image Quality Evaluator (NIQE) [67]. To this end, we generate 1000 images from each of the three RLSO models trained on different prompts, as well as from the vanilla LDM, and report the average score for each metric.

Table 9: Performance of RLSO compared to the vanilla LDM on independent quality metrics

	CLIP ↑		CLIP-IQA ↑		NIQE ↓	
	RLSO	LDM	RLSO	LDM	RLSO	LDM
Prompt A	**36.23**	35.18	0.949	**0.950**	**3.799**	3.864
Prompt B	**36.33**	36.08	0.959	**0.959**	3.398	**3.387**
Prompt C	**35.01**	33.50	0.831	**0.840**	3.363	**3.301**

Prompt A: *A photo of an astronaut riding a horse on mars.* Prompt B: *Yellow cat sitting on a park bench.* Prompt C: *A green colored rabbit.*

The results reported in Table 9 show that RLSO clearly improves prompt faithfulness (CLIP score), while the minor differences in general image quality metrics (CLIP-IQA, NIQE) indicate that RLSO doesn't degrade general image quality.

To evaluate diversity, we generate 1000 images with RLSO and 10000 images with the LDM. We encode both generated datasets with CLIP and measure the mean pairwise cosine similarity δ between encodings within each dataset. We report $\delta = 0.8457$ for the LDM and $\delta = 0.8857$ for RLSO. Furthermore, we investigate how well RLSO trained on a fixed prompt generalizes to unseen prompts and slight architecture modifications. Figures 8 (a,b,c) show that RLSO retains some effectiveness on unseen prompts, although with substantially reduced improvements in sample quality. Similarly, it can cope with dropping the Hyper-SD LoRA at inference time, albeit with slightly reduced effectiveness.

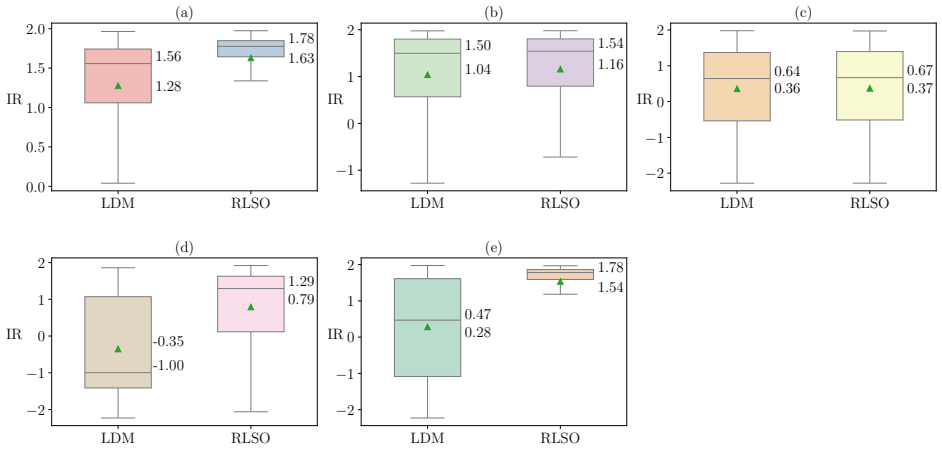


Figure 8: (a) Training with the fixed prompt *"A photo of an astronaut riding a horse on mars."*, reward evaluation with the same fixed prompt.
(b) Training with the fixed prompt *"A photo of an astronaut riding a horse on mars."*, reward evaluation with fixed prompt *yellow cat sitting on a park bench*.
(c) Training with the fixed prompt *"A photo of an astronaut riding a horse on mars."*, reward evaluation with dynamic prompt, sampled at each inference step from the DrawBench dataset [75].
(d) Training with the fixed prompt *"A photo of an astronaut riding a horse on mars."*, reward evaluation with the same fixed prompt, without the Hyper-SD LoRA.
(e) Training with the fixed prompt *"A green colored rabbit."*, reward evaluation with the same fixed prompt. IR denotes the ImageReward score.

E Implementation Details

E.1 Training

Table 10: Hardware details and approximate training time to convergence (one training step comprises one full denoising pass of the frozen LDM).

Variant	GPU	Training steps	Hours
Continuous - Darcy Flow	1x Nvidia GeForce RTX 4090	2×10^6	$\sim$14
Continuous - Voronoi	1x Nvidia GeForce RTX 4090	1.5×10^6	$\sim$11
Discrete - Darcy Flow	1x Nvidia GeForce RTX 4090	0.6×10^6	$\sim$5.2
Discrete - Voronoi	1x Nvidia GeForce RTX 4090	0.8×10^6	$\sim$6.7
Discrete - Stable Diffusion	1x Nvidia GeForce RTX 5090	1.5×10^6	$\sim$54

E.2 Hyperparameters

Table 11: Diffusion model architecture. All other unspecified hyperparameters follow the default Unconditional LDM implementation of Rombach et al. [2].

Hyperparameter	Value
In-, output channels (Darcy Flow)	$2, 2$
In-, output channels (Voronoi)	$1, 1$
In-, output channels (Stable Diffusion)	$4, 4$
ResNet blocks per down-/up-sampling	2
ResNet block normalization	Group Normalization
ResNet block activation function	SiLU
Attention block normalization	LayerNorm
Feature map resolutions	$[64, 64, 128, 256]$
Attention head dimension	32

Table 12: PPO hyperparameters. All other unspecified hyperparameters follow the default PPO implementation of Raffin et al. [76].

Hyperparameter	Value
Actor and Critic Hidden layers	$[2048, 2048]$
PPO Epochs per Update	10
Batch Size	128
Rollout Length	256
Parallel Environments	48
Policy Clip Range	0.2
Value Function Clip Range	0.2
Entropy Coefficient	0.04
Value Loss Coefficient	1.0
Dropout Rate	0.3
Learning Rate	Linear Schedule $[1 \cdot 10^{-6}, 1 \cdot 10^{-7}]$
GAE Lambda	0.9

Table 13: VAE architecture. All other unspecified hyperparameters follow the default AutoencoderKL implementation of von Platen et al. [55].

Hyperparameter	Value
Latent channels (Darcy Flow)	2
Latent channels (Voronoi)	1
Latent channels (Stable Diffusion)	4
Latent dimension (unconditional LDM)	16×16
Latent dimension (Stable Diffusion)	64×64
Down block type, number	DownEncoderBlock2D, 3
Up block type, number	UpDecoderBlock2D, 3
Block output channels	$[64, 64, 64]$

F Qualitative Results

Figures 9 to 13 show a qualitative comparison between the LDM and RLSO. For each comparison, we generated 1000 samples with the LDM and 1000 samples with RLSO and respectively selected 5 at random using a pseudo-random number generator.

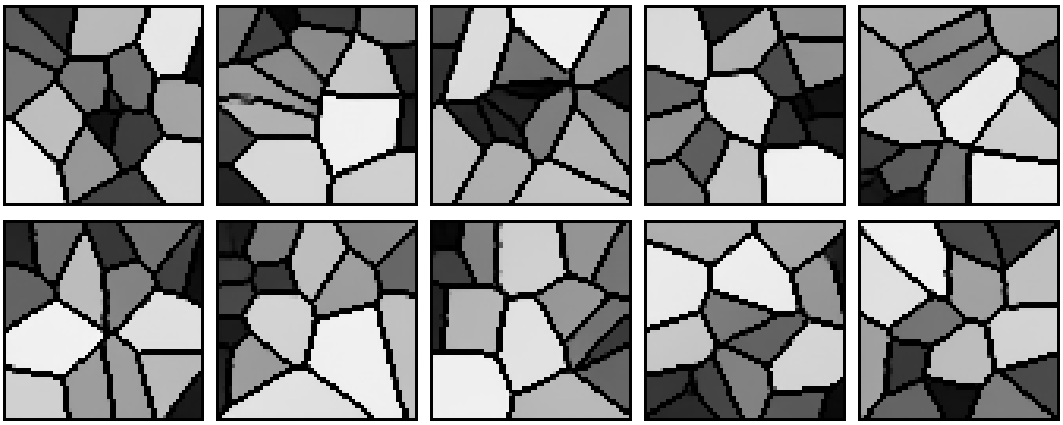


Figure 9: Voronoi samples generated by the unconditional LDM. The mean residual for the LDM samples is $\varepsilon = 1.879$ (top). Voronoi samples generated by RLSO with the continuous action space. The mean residual for the RLSO samples is $\varepsilon = 1.51$ (bottom).

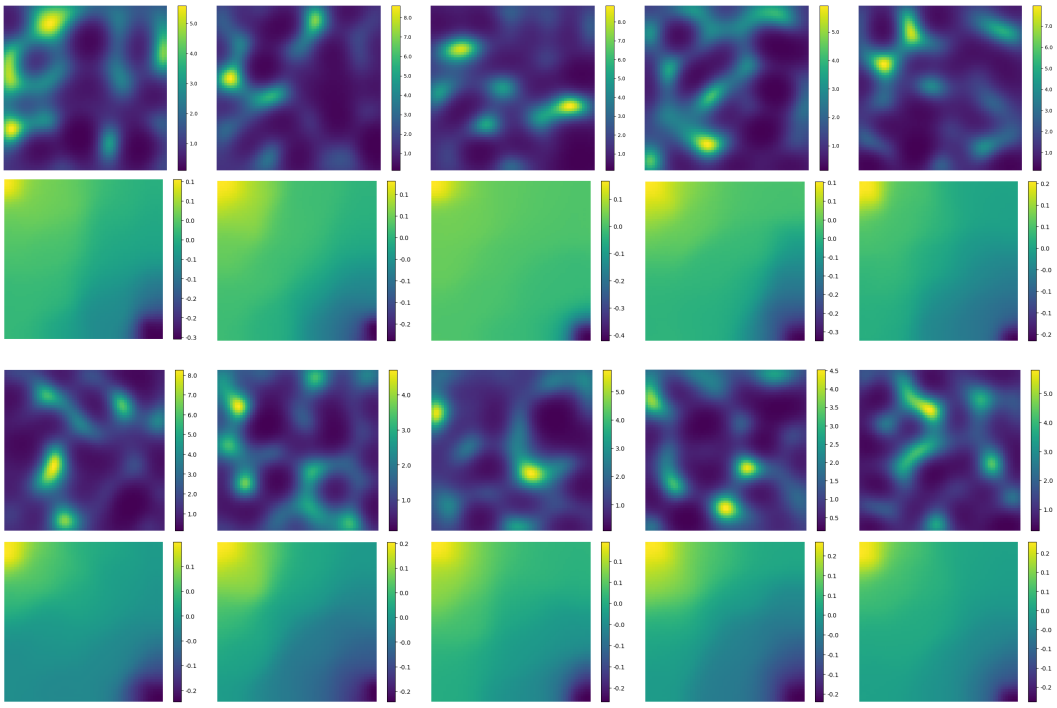


Figure 10: Darcy Flow permeability fields K and pressure fields p generated by the unconditional LDM (top two rows). Row 1 shows samples of K, and Row 2 shows samples of p. The mean residual for the LDM samples is $\varepsilon = 0.552$. Darcy Flow permeability and pressure fields generated by RLSO with the discrete action space ($N_d = 512$) are shown in the bottom two rows. Row 3 shows K, and Row 4 shows p. The mean residual for the RLSO samples is $\varepsilon = 0.424$.



Figure 11: Images generated by the text-conditioned LDM with the prompt *"A photo of an astronaut riding a horse on mars"*: mean reward $r = 1.09$ (top). Images generated by RLSO: mean reward $r = 1.525$ (bottom). Note the absence of a horse in two of the images generated by the LDM.



Figure 12: Images generated by the text-conditioned LDM with the prompt *"Four roses."*: mean reward $r = -1.08$ (top). Images generated by RLSO: mean reward $r = 1.217$ (bottom). Note that RLSO avoids the SD 1.5 failure mode associated with the whisky brand of the same name.



Figure 13: Images generated by the text-conditioned LDM with the prompt *"A green colored rabbit."*: mean reward $r = -0.204$ (top). Images generated by RLSO: mean reward $r = 1.733$ (bottom). Note the presence of a gray and a brown rabbit in the images generated by the LDM.

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 claims made in the abstract and in the introduction fall into two categories: quantitative and descriptive. The quantitative claims are directly extrapolated from the experimental results presented in the main body of the paper; the descriptive ones, using terminology like *significantly outperforms*, *effective*, are logical implications of the quantitative ones.

   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 main body of the paper contains an explicit paragraph titled *Limitations*, where we briefly discuss them. Regarding computational efficiency, the Appendix contains precise Hardware specifications and training times.

   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: [N/A]

   Justification: Our work is grounded in Theory, using and referencing established theorems or formulas from related works. Nevertheless, our paper does not introduce novel Theorems that require formal proofs.

   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: We have provided all information needed to reproduce our experiments, including references to the datasets used and any modifications made, the codebases and libraries used, and all hyperparameters that differ from library defaults. We described our proposed method extensively, including algorithm pseudo-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: [No]

   Justification: While we do not provide code with our submission, we believe that our extensive use of open-source libraries and the extensive description of our method makes our experiments reproducible.

   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: The full relevant details are provided in the Appendix.

   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: Training curves report the mean over 5 random seeds with shaded regions denoting ± 1 standard deviation across seeds, capturing variability from initialization, environment interaction, and stochastic training. Boxplots summarize distributions over 10000 generated samples from a fixed trained model and therefore reflect sample-level variability rather than training-run variability.

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: The full details are provided in the Appendix.

   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: We have reviewed the NeurIPS code of ethics and we believe that our work conforms with it in every respect.

   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 discusses potential positive societal impacts in the introduction, particularly for scientific applications. We do not explicitly discuss negative impacts, as we do not expect our method to introduce significant risks beyond those already associated with the pre-trained diffusion models it improves.

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: Our work uses publicly available datasets and models, and we do not release any new model weights or datasets.

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: All creators of assets used in this paper are appropriately cited. All datasets, code, and weights used were released under licenses which permit their use for research purposes.

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: [N/A]

    Justification: We do not release new assets.

    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 paper does not involve crowdsourcing nor research with human subjects.

    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 paper does not involve crowdsourcing nor research with human subjects.

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 development in this paper does not involve LLMs as any important, original, or non-standard components.

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.