DDNO: DISCRETE DIFFUSION NOISE OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-time reward alignment of discrete diffusion models currently relies on either step-wise guidance, which uses myopic approximations that can distort the generative process, or on search with stochastic generation. We introduce Discrete Diffusion Noise Optimization (DDNO), a training-free method for uniform discrete diffusion models that enables gradient-based reward optimization at test-time despite the non-differentiable categorical sampling that blocks gradient flow through the reverse process. DDNO relaxes the discrete initial noise to continuous logits and propagates terminal reward gradients end-to-end through straight-through estimation and soft mixing. By fixing all intermediate randomness, generation becomes controllable through variation of the initial noise. On text-to-image generation and steering text generation, DDNO outperforms existing test-time methods, achieving 0.94 GenEval overall accuracy, with gains measured under held-out evaluators distinct from the optimization reward. Under matched compute, search-based methods are preferable at smaller budgets but plateau, while DDNO scales consistently with optimization budget across both modalities, establishing gradient-based noise optimization as a new axis for test-time scaling in discrete generative models.

1 INTRODUCTION

Discrete generative models are increasingly used not only to imitate data, but to solve *problems*. We desire discrete sequences that are fluent, faithful, and useful according to a downstream objective. This is formalized as *reward-guided generation*, where a reward function $R(\cdot)$ scores outputs for properties such as constraint satisfaction or human preference (Ouyang et al., 2022; Ziegler et al., 2019). Discrete diffusion models are a natural target: they now power both language models and unified multimodal models that generate text and images in a shared token space (Wang et al., 2025b).

A key insight from continuous diffusion is that initial noise is not merely a source of stochasticity, but a *control variable*: with intermediate randomness fixed, the reverse process becomes a deterministic map from noise to output, and the noise can be optimized directly for a reward (Ben-Hamu et al., 2024; Eyring et al., 2024; Wallace et al., 2023). The same principle applies to uniform discrete diffusion (Austin et al., 2021; Campbell et al., 2024): with fixed intermediate stochasticity, the generated sequence is fully determined by its initial categorical noise. However, categorical sampling is non-differentiable, so gradients from a terminal reward cannot flow back to the initial noise.

Existing reward-alignment methods for discrete diffusion do not exploit this structure. Fine-tuning via reinforcement learning (Zekri & Boullé, 2025; Zhao et al., 2025) or preference optimization (Borso et al., 2025) is expensive and produces specialized checkpoints. Training-free guidance (Nisonoff et al., 2024; Schiff et al., 2025) modifies the denoising process through per-step corrections, which can distort the generative process when pushed aggressively. Search-based methods, from Best-of-N to sequential Monte Carlo (Li et al., 2024a; Singhal et al., 2025), explore the model's outputs by sampling and selection: they improve quickly at small budgets, but their gains saturate as compute grows and vanish when high-reward outputs are rare under the base model.

We introduce **Discrete Diffusion Noise Optimization (DDNO)**, a test-time method that makes initial discrete noise optimizable. DDNO relaxes the initial categorical noise to continuous logits $\boldsymbol{\ell}_0$, runs the frozen reverse process with fixed intermediate randomness, and updates $\boldsymbol{\ell}_0$ by gradient ascent on the terminal reward. To make this possible, we construct a surrogate that keeps sampled tokens discrete in the forward pass while propagating gradients end-to-end via straight-through estimation and soft mixing (Maddison et al., 2016). Compared to search-based methods, DDNO turns random trajectory sampling into gradient-guided trajectory search (Figure 1).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

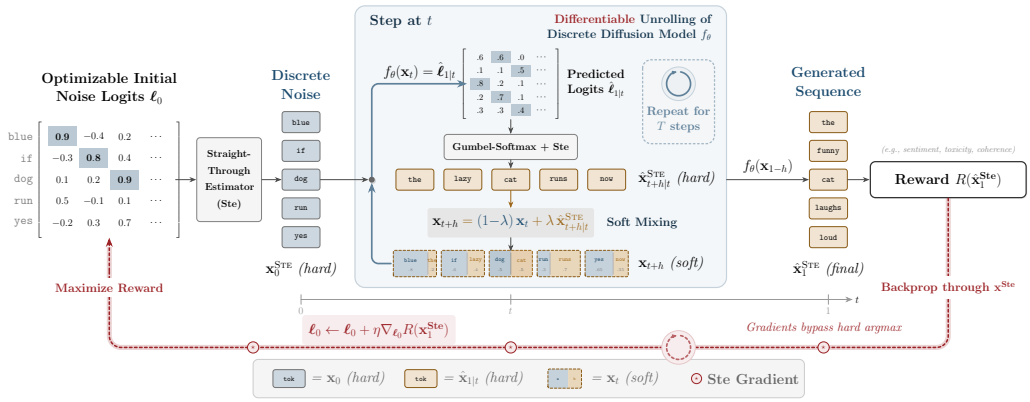

Figure 1: **Discrete Diffusion Noise Optimization (DDNO).** From left to right: optimizable noise logits ℓ_0 parameterize the initial discrete noise fed to a frozen uniform discrete diffusion model f_θ. The reverse process is unrolled differentiably via GUMBELSTE and soft mixing (Equation (12)), and a terminal reward R drives gradient ascent back to ℓ_0 (Equation (10)).

We benchmark DDNO on text-to-image generation and on controllable text generation with held-out evaluators throughout. We find that search-based methods are preferable to DDNO at low budgets but plateau, while DDNO keeps improving and overtakes them with more compute, both for images and text, outperforming all competing test-time methods at higher budgets. Our main contributions are:

1. **Initial noise as control:** We introduce DDNO, a test-time scaling framework for reward alignment through optimizing initial noise in discrete diffusion models.
2. **Differentiable surrogate:** We develop a differentiable surrogate combining GUMBELSTE sampling with soft mixing that closely approximates discrete dynamics while enabling gradient flow.
3. **Test-time scaling:** We show that DDNO defines a gradient-based scaling axis that outscales search-based methods under matched compute on both compositional image generation (0.94 GenEval) and text generation, with consistent gains as optimization budget increases.

2 DDNO: DISCRETE DIFFUSION NOISE OPTIMIZATION

Notation. We consider sequences $\mathbf{x} = (x^1, \ldots, x^D) \in \mathcal{S} = \mathcal{T}^D$ over a token alphabet $\mathcal{T} = \{1, \ldots, K\}$. Let $p_{\text{data}}(\mathbf{x})$ denote the data distribution and $p_{\text{noise}}(\mathbf{x})$ the source distribution. Time $t \in [0, 1]$ indexes a probability path $\{q_t\}$. For tokens, $\delta_a(x) = \mathbf{1}\{x = a\}$. We write $\bar{x}_t^i \in \mathcal{T}$ for the hard (discrete) token at position i, $\bar{\mathbf{x}}_t^i \in \{0, 1\}^K$ for its one-hot encoding, and $\tilde{\mathbf{x}}_t^i \in \Delta^{K-1}$ for its soft (probability simplex) representation.

2.1 BACKGROUND

Discrete Diffusion and Flow Matching construct a continuous-time path of distributions interpolating between a tractable source $p_{\text{noise}}(\mathbf{x})$ and the data distribution $p_{\text{data}}(\mathbf{x})$ such that

$$q_0(\mathbf{x}) = p_{\text{noise}}(\mathbf{x}), \qquad q_1(\mathbf{x}) = p_{\text{data}}(\mathbf{x}). \tag{1}$$

Paths are built by conditioning on clean data $\mathbf{x}_1 \sim p_{\text{data}}$ with a factorized interpolation. The canonical choice is a convex mixture with scheduler $\alpha_t \in [0, 1]$

$$q_t(x^i | x_1^i) = (1 - \alpha_t) p_{\text{noise}}(x^i) + \alpha_t \delta_{x_1^i}(x^i), \quad \alpha_0 = 0, \alpha_1 = 1. \tag{2}$$

Uniform Prior. Throughout this work we use a uniform noise prior, $p_{\text{noise}}(x) = K^{-1}$ per token. Under uniform noise, different initial sequences $\mathbf{x}_0$ traverse different trajectories through the generative process, yielding different samples. DDNO exploits this by optimizing *which* initial sequence to start from. Masked diffusion, by contrast, initializes every trajectory from the same deterministic state (all `[MASK]` tokens), leaving no variation in initial noise to optimize. While masked diffusion has been

the predominant paradigm, recent work demonstrates that uniform-noise models can match or exceed their performance (DiffusionGemmaTeam et al., 2026; Sahoo et al., 2025; Schiff et al., 2025; Shaul et al., 2024; von Rütte et al., 2025), making this assumption increasingly practical.

Learning. The interpolation induces a CTMC governed by a rate matrix (Campbell et al., 2024; Lipman et al., 2024) (details in Section A.2). A neural network f_θ predicts the posterior over clean tokens given noisy states:

$$\hat{q}_{1|t}(x_1^i = k \mid \mathbf{x}_t) = \mathrm{softmax}(f_\theta(\mathbf{x}_t, t)^i)_k. \tag{3}$$

Sampling. Generation simulates the CTMC from $t{=}0$ (noise) to $t{=}1$ (data). At each step $t \to s$, the posterior prediction combines with the rate-matrix structure to yield transition probabilities $q^i_{s|t}(\cdot \mid \mathbf{x}_t)$. We consider two strategies. **Ancestral sampling** uses a *jump/stay* decomposition: for current token $c = \bar{x}_t^i$,

$$p^i_{\mathrm{jump}} \coloneqq 1 - q^i_{s|t}(c \mid \mathbf{x}_t), \quad r^i_{s|t}(k) \coloneqq \frac{q^i_{s|t}(k \mid \mathbf{x}_t)\,\mathbf{1}\{k \neq c\}}{p^i_{\mathrm{jump}}} \quad (\text{when } p^i_{\mathrm{jump}} > 0). \tag{4}$$

The sampler draws a destination $y^i \sim \mathrm{Cat}(r^i_{s|t})$ and mask $m^i \sim \mathrm{Bernoulli}(p^i_{\mathrm{jump}})$, setting $\bar{x}_s^i = c$ if $m^i = 0$ and $\bar{x}_s^i = y^i$ otherwise. **Adaptive sampling** (von Rütte et al., 2025) deterministically selects positions where the model is most confident a change is needed. Each position receives a score

$$\sigma^i = (p^i_{\mathrm{best}} - p^i_{\mathrm{curr}})\,\pi_\lambda(\bar{x}_t^i), \qquad p^i_{\mathrm{best}} = \max_k \hat{q}_{1|t}(k \mid \mathbf{x}_t), \quad p^i_{\mathrm{curr}} = \hat{q}_{1|t}(\bar{x}_t^i \mid \mathbf{x}_t), \tag{5}$$

where $\pi_\lambda(\bar{x}_t^i)$ is the prior probability of the current token. The top-k positions by score are overwritten with tokens sampled from $\hat{q}_{1|t}(\cdot \mid \mathbf{x}_t)$.

The Gradient Barrier. Across positions, these discrete update rules define a generative map whose categorical sampling operations block gradient propagation from a terminal reward $R(\mathbf{x}_1)$. Existing guidance methods (Nisonoff et al., 2025; Schiff et al., 2025) address this by modifying transition rates at every step, but rely on approximations that can degrade quality (Hasan et al., 2026; Lee et al., 2025; Wan et al., 2025); see Section A.4 for a formal treatment. This motivates optimizing the *source* of the generative process rather than perturbing its dynamics.

2.2 Optimization Objective

We introduce **Discrete Diffusion Noise Optimization (DDNO)**, a training-free framework for inference-time reward alignment of discrete diffusion models. The central idea is to treat the initial noise sequence as a *control variable* optimized to steer generation toward high-reward outputs.

Let $G_\theta(\cdot; \xi)$ denote the reverse sampler, where ξ collects the intermediate random variables used by categorical sampling. For a fixed ξ, the sampler maps an initial state $\mathbf{x}_0$ to a terminal sample $\mathbf{x}_1$. Rather than perturbing the dynamics through step-wise guidance, we seek initial noise sequences $\mathbf{x}_0^\star$ from which the *unchanged* model G_θ naturally evolves toward high-reward sequences. Sections 2.3 and 2.4 construct a surrogate in which ξ is fixed rather than resampled, converting the stochastic sampler into a deterministic, differentiable map amenable to gradient ascent.

Per-sample objective. Let $V(\mathbf{x}_0) \coloneqq \mathbb{E}_\xi[R(G_\theta(\mathbf{x}_0; \xi))]$ denote the expected terminal reward when the frozen reverse process is initialized at $\mathbf{x}_0$. We seek the initial sequence that maximizes this value:

$$\mathbf{x}_0^\star \in \arg\max_{\mathbf{x}_0 \in \mathcal{S}} \; V(\mathbf{x}_0). \tag{6}$$

This is a mode-seeking objective: we want a single high-reward output, not a sample from a distribution. A distributional generalization with KL regularization toward the noise prior can be derived from a reward-tilting perspective (Section A.4); the per-sample formulation equation 6 arises as its Dirac restriction under the uniform prior (Lemma A.4, Corollary A.5).

Optimization challenges. Equation (6) specifies *what* to find; the difficulty is *how*. The search space $\mathcal{S} = \{1, \ldots, K\}^D$ is discrete and combinatorially large. Moreover, G_θ involves categorical sampling at every step, blocking gradient flow from the terminal reward. The following sections develop the necessary machinery: a continuous parametrization of the discrete initial noise (Section 2.3), and a differentiable deterministic surrogate for the reverse dynamics (Section 2.4).

Algorithm 1 Discrete Diffusion Sampling: ***Standard*** vs. ***Differentiable Surrogate***

Require: Model f_θ, steps $1/h$, Vocab K, length D, Temp τ

Initialize:

Standard Prior	***Differentiable Init***
$\mathbf{x}_0 \sim \mathrm{Uniform}([K])^{\otimes L}$	$\boldsymbol{\ell}_0 \sim \mathcal{N}(0, \mathbf{I}), \quad \mathbf{x}_0 \leftarrow \textsc{Ste}(\boldsymbol{\ell}_0, \tau)$

for $t = 0, h, 2h, \ldots, 1-h \quad (\text{let } s = t + h)$ **do**

Predict and compute transition: *// Shared backbone*

$\hat{\boldsymbol{\ell}}_{1|t} \leftarrow f_\theta(\mathbf{x}_t, t), \quad \mathbf{p}_{\text{jump}}, \mathbf{r}_{s|t} \leftarrow \textsc{Transition}(\mathbf{x}_t, \hat{\boldsymbol{\ell}}_{1|t}, t, s)$ // Eq. 4

Next State Transition:

Categorical Sampling	***Differentiable Sampling***
$\mathbf{x}_{s\|t} \sim \mathrm{Cat}(\mathbf{r}_{s\|t}), \quad \mathbf{m} \sim \mathrm{Bernoulli}(\mathbf{p}_{\text{jump}})$	$\mathbf{x}_{s\|t} \leftarrow \textsc{GumbelSte}(\log \mathbf{r}_{s\|t}, \tau)$
Hard Mixing	***Soft Mixing***
$\mathbf{x}_s \leftarrow (\mathbf{1} - \mathbf{m}) \odot \mathbf{x}_t + \mathbf{m} \odot \mathbf{x}_{s\|t}$	$\mathbf{x}_s \leftarrow (\mathbf{1} - \mathbf{p}_{\text{jump}}) \odot \mathbf{x}_t + \mathbf{p}_{\text{jump}} \odot \mathbf{x}_{s\|t}$

end for

Return: $\mathbf{x}_1$

2.3 From Discrete Noise to Optimizable Logits

The initial state $\mathbf{x}_0$ is discrete, precluding direct gradient-based optimization. We introduce a continuous parameterization that enables gradient flow, i.e. we parametrize the initial discrete state $\mathbf{x}_0$ through continuous logits $\boldsymbol{\ell}_0 \in \mathbb{R}^{D \times K}$, which implicitly define a factorized categorical distribution

$$q_{0,\boldsymbol{\ell}_0}(\mathbf{x}_0) = \mathrm{Cat}(\mathbf{x}_0 \mid \mathrm{softmax}(\boldsymbol{\ell}_0)). \tag{7}$$

Hard sampling via (Gumbel) straight-through estimation. To propagate reward gradients through discrete sampling, we use a straight-through surrogate with hard tokens in the forward pass and a continuous relaxation in the backward pass. For logits $\boldsymbol{\ell} \in \mathbb{R}^K$ and temperature $\tau > 0$, we leverage the (Gumbel) straight-through estimator (Jang et al., 2016; Maddison et al., 2016)

$$\boldsymbol{\epsilon} \sim \mathrm{Gumbel}(0,1)^K, \quad \mathbf{p}_{\boldsymbol{\epsilon}} = \mathrm{softmax}\big((\boldsymbol{\ell} + \boldsymbol{\epsilon})/\tau\big), \qquad \mathbf{h}_{\boldsymbol{\epsilon}} = \mathrm{onehot}(\arg\max_k(\ell_k + \epsilon_k)), \tag{8}$$

$$\textsc{GumbelSte}(\boldsymbol{\ell}, \tau) = \mathbf{p}_{\boldsymbol{\epsilon}} + \mathrm{sg}(\mathbf{h}_{\boldsymbol{\epsilon}} - \mathbf{p}_{\boldsymbol{\epsilon}}), \tag{9}$$

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator. Setting $\boldsymbol{\epsilon} = \mathbf{0}$ recovers the deterministic straight-through estimator, $\textsc{Ste}(\boldsymbol{\ell}, \tau) \coloneqq \textsc{GumbelSte}(\boldsymbol{\ell}, \tau)\big|_{\boldsymbol{\epsilon}=\mathbf{0}}$ (Bengio et al., 2013).

Initialization. The pretrained model assumes uniform noise $p_{\text{noise}}(\mathbf{x}) = K^{-D}$. Zero logits yield uniform softmax probabilities, but their argmax is degenerate: ties would map every position to the same token, so the initial state would not be a sample from the prior. Instead, we initialize $\ell_0^{i,k} \sim \mathcal{N}(0,1)$ and compute $\mathbf{x}_0$ from it through $\arg\max$. By exchangeability of Gaussian random variables, $\arg\max_k \ell_0^{i,k}$ is exactly uniform over $\{1, \ldots, K\}$, matching the prior.

Now instead of standard sampling $\mathbf{x}_0 \sim \mathrm{Uniform}([K])^{\otimes D}$, we obtain the initial discrete sequence by first sampling Gaussian logits $\boldsymbol{\ell}_0 \sim \mathcal{N}(0, \mathbf{I})$ and then applying $\mathbf{x}_0 = \textsc{Ste}(\boldsymbol{\ell}_0, \tau)$. This enables gradient-based optimization of the initial noise through its continuous parametrization $\boldsymbol{\ell}_0$. Note that injecting Gumbel noise at initialization is not needed here and would only increase estimator variance; we reserve GumbelSte for categorical sampling inside the unrolled reverse process (Section 2.4).

Objective. The logits $\boldsymbol{\ell}_0$ implicitly define a factorized categorical $q_{0,\boldsymbol{\ell}_0}(\cdot)$ through which gradients flow via Ste. We regularize this distribution toward the uniform prior by maximizing its entropy, which instantiates the variational regularizer from Equation (27) and prevents logit saturation that would degrade Ste gradients. Let $\widetilde{G}_\theta$ denote the differentiable deterministic surrogate defined in Section 2.4. The full DDNO test-time objective reads

$$\boldsymbol{\ell}_0^\star = \arg\max_{\boldsymbol{\ell}_0} \; R\big(\widetilde{G}_\theta(\textsc{Ste}(\boldsymbol{\ell}_0, \tau))\big) + \frac{1}{\beta} \sum\nolimits_{i=1}^{D} H\big(\mathrm{softmax}(\boldsymbol{\ell}_0^i)\big), \tag{10}$$

where $H(\cdot)$ denotes entropy and β controls regularization strength. We optimize this via gradient ascent on $\boldsymbol{\ell}_0$. The logit parameterization and entropy regularization are *sampler-agnostic*: they apply independently of how the reverse process advances. The next section constructs $\widetilde{G}_\theta$ by making the stochastic reverse dynamics both deterministic and differentiable.

2.4 DETERMINISTIC DIFFERENTIABLE SURROGATE

The key step in DDNO is to make the stochastic reverse process pathwise deterministic during optimization. The surrogate leverages Gumbel noise $\boldsymbol{\epsilon}_t^i$ at every denoising step. If these samples are redrawn at each gradient step, identical logits $\boldsymbol{\ell}_0$ can produce vastly different trajectories and rewards, creating a noisy optimization landscape. We eliminate this source of variance by sampling *all* Gumbel vectors $\{\boldsymbol{\epsilon}_t^i\}_{t,i}$ once at initialization and holding them fixed throughout optimization, rendering the surrogate a **deterministic map** $\boldsymbol{\ell}_0 \mapsto \mathbf{x}_1(\boldsymbol{\ell}_0; \{\boldsymbol{\epsilon}_t^i\})$.

Crucially, setting $\boldsymbol{\epsilon} = \mathbf{0}$ everywhere—recovering a pure STE throughout the reverse process—is not equivalent. Without stochastic perturbations, positions that share similar model logits break ties identically at every step. The fixed-but-nonzero strategy preserves the per-position diversity of categorical sampling while providing a smooth, reproducible loss landscape for gradient ascent.

The reverse process G_θ still involves two sources of non-differentiability: (i) categorical sampling of destination tokens, and (ii) a position-update rule, where in the *ancestral* case, binary jump decisions select whether each position updates. We address (i) with Gumbel straight-through estimation and (ii) with a soft mixing scheme that replaces stochastic masks with their conditional expectations.

Differentiable Sampling. We replace each non-differentiable categorical draw $\mathbf{x}_{s|t} \sim \mathrm{Cat}(\mathbf{r}_{s|t})$ with its differentiable counterpart $\mathbf{x}_{s|t} \leftarrow \textsc{GumbelSte}(\log \mathbf{r}_{s|t}, \tau)$, using the fixed Gumbel vectors $\boldsymbol{\epsilon}_t^i$ described above. The forward pass produces *discrete tokens* (via the argmax in the straight-through estimator), while the backward pass uses the *soft* Gumbel-Softmax gradients, enabling end-to-end gradient flow through the entire sampling chain.

Soft Mixing for Ancestral Sampling. As described in Equation (4), the standard sampler uses a two-stage procedure: sample a destination $\mathbf{x}_{s|t} \sim \mathrm{Cat}(\mathbf{r}_{s|t})$, then apply a binary mask $\mathbf{m} \sim \mathrm{Bernoulli}(\mathbf{p}_{\mathrm{jump}})$ to decide whether each position jumps

$$\mathbf{x}_s = (\mathbf{1} - \mathbf{m}) \odot \mathbf{x}_t + \mathbf{m} \odot \mathbf{x}_{s|t}. \tag{11}$$

The binary mask $\mathbf{m}$ blocks gradient flow: positions that stay ($m^i = 0$) receive no gradient signal from downstream losses. To restore differentiability, we replace the stochastic mask with its expectation

$$\mathbf{x}_s = (\mathbf{1} - \mathbf{p}_{\mathrm{jump}}) \odot \mathbf{x}_t + \mathbf{p}_{\mathrm{jump}} \odot \mathbf{x}_{s|t}. \tag{12}$$

This soft mixing interpolates between exactly two tokens per position—the current state $\mathbf{x}_t$ and the sampled destination $\mathbf{x}_{s|t}$—using continuous weights $\mathbf{p}_{\mathrm{jump}}$ that carry gradient to the denoiser. This substitution is an instance of Rao-Blackwellization (Casella & Robert, 1996; Robert & Roberts, 2021): conditioned on the sampled destination $\mathbf{x}_{s|t}$, we compute the conditional expectation $\mathbb{E}[\mathbf{x}_s \mid \mathbf{x}_{s|t}]$ rather than sampling the mask. Since $\mathbf{m} \perp \mathbf{x}_{s|t}$, the law of total variance guarantees this preserves the mean while reducing variance at the state-estimator level (Appendix B.1). We summarize our approach in Algorithm 1 and Figure 1.

DDNO with Adaptive Sampling. The adaptive sampler (Equation (5)) selects positions deterministically via TopK and overwrites them with the model's token proposal. Unlike in ancestral sampling, there is no stochastic mask—the position-update rule is a deterministic overwrite at selected positions

$$\mathbf{x}_s = \mathbf{x}_t + \mathrm{sg}(\mathbf{U}_t) \odot (\mathbf{x}_{\mathrm{prop}} - \mathbf{x}_t), \qquad U_t^i = \mathbf{1}[i \in \mathcal{I}_t], \tag{13}$$

where $\mathcal{I}_t = \mathrm{TopK}(\mathrm{sg}(\sigma_t), m)$ are the m positions with highest adaptive score and $\mathbf{x}_{\mathrm{prop}}$ is the GUMBELSTE token proposal. The stop-gradient operator sg blocks gradient flow through the discrete position selection but not through the token values such that gradients naturally propagate through $\mathbf{x}_{\mathrm{prop}}$ at selected positions and through $\mathbf{x}_t$ at the remaining ones. Algorithm 2 details this.

3 EXPERIMENTS

Our experiments address three main questions: **(i)** Can a differentiable surrogate faithfully approximate discrete sampling while enabling gradient flow? **(ii)** Does optimizing the initial discrete noise outperform existing inference-time methods? **(iii)** Does DDNO generalize across modalities?

We evaluate DDNO on compositional text-to-image generation, as well as text topic steering and reasoning tasks. Throughout, model parameters remain frozen; only initial noise logits are optimized. The main hyperparameters of DDNO are the chosen learning rate and iteration steps (Algorithm 3).

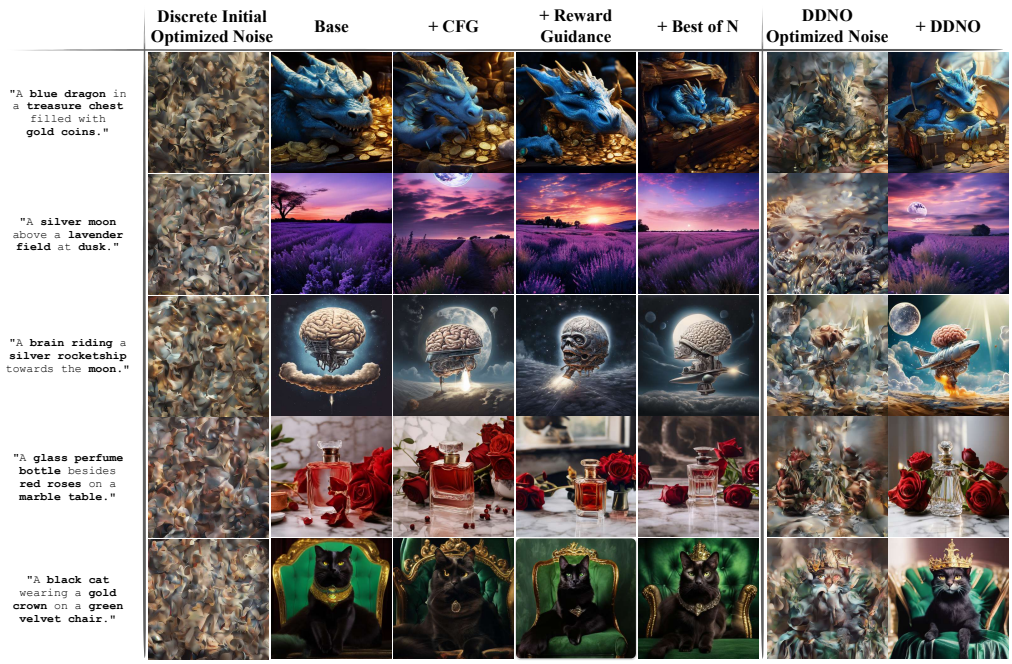

Figure 2: **Discrete Diffusion Noise Optimization (DDNO).** Qualitative comparison across five prompts for a frozen discrete diffusion model. **Left:** Existing inference-time methods (CFG, reward guidance, Best-of-N) applied to randomly sampled discrete initial noise. **Right:** DDNO optimized noise produces generations with improved prompt alignment and visual quality. Notably, the *DDNO Optimized Noise* column reveals that DDNO embeds coarse semantic structure directly into the noise.

Image generation setup. We leverage FUDOKI (Wang et al., 2025b), a discrete flow matching model with kinetic-optimal paths (Shaul et al., 2024), and evaluate on GenEval (Ghosh et al., 2023), which measures compositional understanding across six categories. We use a combination of an aesthetic quality reward model (`HPSv2.1`) and `NVILA-Lite-2B-Verifier`, a 2B fine-tuned VLM model to check for prompt following measured through the probability of answering `yes` to a prompt following question: $R = R_{\text{fidelity}} + R_{\text{aesthetic}}$. We optimize noise logits for M steps, defaulting to 64, using Adam with learning rate 0.1. We compare against CFG (Ho & Salimans, 2022), reward guidance (Nisonoff et al., 2025), for which we sweep guidance strength for all experiments, Best-of-N sampling (Karthik et al., 2023; Ma et al., 2025), SVDD-PM (Li et al., 2024a), and FK-Steering (Singhal et al., 2025). We report the full implementation details in Section F.

3.1 Surrogate Design Ablation

We ablate the surrogate along three axes: (i) *mixing*, soft (Equation (12)) vs. hard Bernoulli masks (Equation (11)); (ii) *sampling*, soft probabilities vs. hard tokens via straight-through estimation; and (iii) *Gumbel noise*, fixed across optimization steps, freshly resampled, or absent (recovering a pure STE). A good surrogate should both faithfully approximate the original sampler and provide stable gradients.

Table 1: **Ablation of surrogate.** *No Opt.*: Surrogate fidelity measured by GenEval at 50 NFE; *+DDNO*: GenEval accuracy after 64 optimization steps. *Soft mixing* with *hard sampling* and *fixed Gumbel* noise achieves faithful generation and stable optimization.

Mixing	Sampling	Gumbel	No Opt.	+DDNO	Base Fid.	Stable Opt.
Hard	Soft	Fixed	0.44	–	✗	–
Soft	Soft	Fixed	0.59	–	✗	–
Hard	Hard	*None*	0.47	–	✗	–
Hard	Hard	Fixed	**0.66**	0.84	✓	✗
Soft	Hard	*Fresh*	0.65	0.85	✓	✗
Soft	**Hard**	**Fixed**	0.65	**0.94**	✓	✓

Table 2: **GenEval results for compositional text-to-image synthesis.** *Top (grey):* Continuous models for reference only. *Bottom:* Test-time methods applied to the same frozen FUDOKI model. CFG and reward guidance use a single sample while the other methods are compute-matched (∼26 GPU-h in total). **Bold** marks the best result among FUDOKI-based methods. The last row decodes the DDNO-optimized noise with the original (non-differentiable) FUDOKI sampler.

Model	Mean↑	Single↑	Two↑	Counting↑	Colors↑	Position↑	Attribution↑
Reference: Continuous Diffusion models							
SD v2.1 (Rombach et al., 2022)	0.50	0.98	0.51	0.44	0.85	0.07	0.17
SD-Turbo (Sauer et al., 2023)	0.49	0.99	0.51	0.38	0.85	0.07	0.14
SDXL (Podell et al., 2023)	0.55	0.98	0.74	0.39	0.85	0.15	0.23
DPO-SDXL (Wallace et al., 2024)	0.59	0.99	0.84	0.49	0.87	0.13	0.24
Flux-dev	0.68	0.99	0.85	0.74	0.79	0.21	0.48
SD3-Medium (Esser et al., 2024)	0.70	1.00	0.90	0.72	0.87	0.31	0.66
Qwen-Image-27B (Wu et al., 2025)	0.87	0.99	0.92	0.89	0.88	0.76	0.77
Discrete Diffusion models + test-time methods							
FUDOKI Base (Wang et al., 2025b)	0.67	0.95	0.73	0.47	0.83	0.55	0.48
+ Reward Guidance (Nisonoff et al., 2025)	0.75	0.97	0.78	0.64	0.87	0.58	0.63
+ CFG (Ho & Salimans, 2022)	0.77	0.93	0.87	0.57	0.90	0.70	0.67
+ Best-of-N (Karthik et al., 2023)	0.91	0.99	0.96	0.83	0.94	**0.92**	0.83
+ SVDD-PM (Li et al., 2024a)	0.90	0.99	0.95	0.81	0.93	0.89	0.83
+ FK-Steering (Singhal et al., 2025)	0.92	0.99	0.96	0.85	0.95	0.91	0.87
FUDOKI Differentiable Surrogate	0.65	0.96	0.70	0.42	0.81	0.53	0.46
+ DDNO (Ours)	**0.94**	**1.00**	**0.98**	**0.89**	**0.98**	0.87	**0.91**
+ DDNO, base decode	0.86	0.99	0.96	0.76	0.93	0.77	0.73

Table 1 reports GenEval scores on FUDOKI (Wang et al., 2025b), where the base model scores 0.67. The *No Opt.* column measures surrogate fidelity without reward optimization. Soft sampling destroys fidelity regardless of mixing strategy (0.44–0.59), confirming that the model requires discrete tokens. Removing Gumbel noise entirely (pure STE) also degrades fidelity (0.47), as $\arg\max$ across positions collapses diversity. Hard mixing with fixed Gumbel achieves the highest base fidelity.

The *+DDNO* column reveals that fidelity alone is insufficient. Hard mixing preserves the highest base fidelity but yields unstable optimization, suggesting that soft mixing is critical for gradient flow through the sampling chain. Our proposed design, combining soft mixing, hard sampling, and fixed Gumbel, achieves stable optimization and the highest post-optimization score (0.94). Using fresh Gumbel noise instead introduces optimization stochasticity that limits performance to 0.85. We ablate the effect of KL regularization in Section E.2 without which DDNO performs marginally worse.

3.2 Image Generation: General Preference Alignment

Table 2 presents the main results. DDNO achieves a mean GenEval score of 0.94, improving over the original FUDOKI sampler (0.67) by 27 pp and over the unoptimized surrogate (0.65) by 29 pp, which indicates that the frozen model already possesses compositional capabilities that a favorable noise initialization can surface. The gains concentrate in the categories where the base model struggles most: counting (+42 pp), position (+32 pp), and attribute binding (+43 pp). Since one forward–backward pass through the surrogate costs about 3× a forward generation, we compare against search-based methods at matched compute, running DDNO with 16 NFE per surrogate pass (instead of 50) and varying its number of steps. We provide a detailed discussion of compute matching in Section E.3. At low budgets ($N \leq 32$), search-based methods are preferable, led by FK-Steering (Figure 3). DDNO, however, scales more efficiently and surpasses all of them at higher budgets, reaching 0.94 at the largest budget versus 0.92 for FK-Steering, 0.91 for Best-of-N, and 0.90 for SVDD-PM. Figure 2 shows that the optimized noise

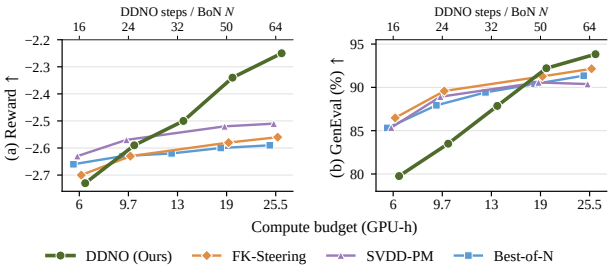

Figure 3: **Compute-matched GenEval scores:** Search-based methods are preferable at low compute budgets, but DDNO scales more efficiently, surpassing them at higher N for both the optimized reward *and* held-out GenEval.

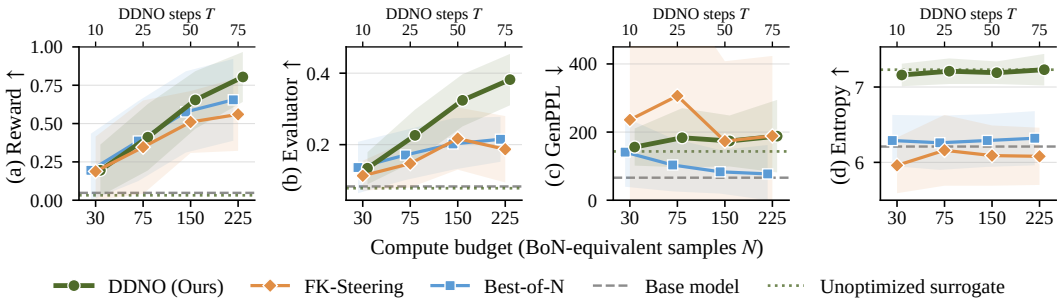

Figure 5: **Compute-matched scaling on topic steering.** DDNO vs. Best-of-N and FK-Steering: (a) optimization reward, (b) held-out evaluator, (c) GenPPL, and (d) sequence entropy. Dotted: unoptimized surrogate; dashed: base sampler; shaded: ± 1 std.

encodes coarse semantic structure, and Section 3.3 confirms the scaling advantage of gradient-based optimization over search-based is consistent across modalities.

Transferability. A natural concern is whether DDNO exploits surrogate-specific artifacts rather than discovering genuinely favorable noise configurations. To test this, we decode surrogate-optimized noise using the original, non-differentiable FUDOKI sampler. As shown in Table 2, the transferred noise achieves a GenEval score of 0.86 (+19 pp) confirming that DDNO discovers noise configurations that carry meaningful semantic structure independently of the surrogate. DDNO is also robust to the initial logit scale, which affects only the gradients, not the initial sample as analyzed in Section E.1.

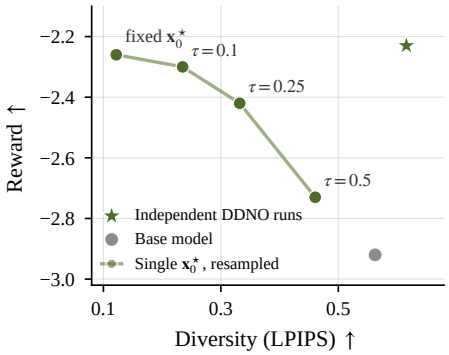

Figure 4: **Diversity–reward trade-off** (4 prompts × 20 samples). Independent runs are more diverse than the base model at higher reward; temperature sampling around one $\mathbf{x}_0^\star$ trades reward for diversity.

Diversity–reward trade-off. We find that the initial noise dominates the output: resampling only the *intermediate noise* around an optimized $\mathbf{x}_0^\star$ preserves the reward (-2.26 vs. -2.23) but yields little diversity (LPIPS 0.12; Figure 4). However, DDNO offers two other sources of diversity. Independent runs converge to distinct modes and are *more* diverse than the base model (LPIPS 0.62 vs. 0.56) at much higher reward (-2.23 vs. -2.92). Unlike continuous noise optimization (Eyring et al., 2024; Wallace et al., 2023), where the optimized noise is a single point, the categorical relaxation in DDNO yields a whole distribution over initial states that can be leveraged to generate multiple distinct initial noise samples. Within a run, sampling $\mathbf{x}_0 \sim \mathrm{Cat}(\mathrm{softmax}(\boldsymbol{\ell}_0^\star/\tau))$ trades reward for diversity smoothly: τ=0.5 recovers 82% of the base diversity while still exceeding the base reward.

3.3 Text Generation

Topic steering. We evaluate whether DDNO transfers beyond images by steering a frozen discrete diffusion language model toward target topics. We use GIDD-Unif-3B (von Rütte et al., 2025), a 3B-parameter uniform-noise discrete diffusion model with adaptive sampling. The reward is a DeBERTa-v3-Large NLI classifier (He et al., 2021; Laurer et al., 2024; Williams et al., 2018) scoring the entailment probability of the hypothesis "This text is about {topic}." To ensure disentangled evaluation, we additionally report a held-out evaluator (RoBERTa-Large-MNLI (Liu et al., 2019)) and GenPPL (GPT2-XL). We use two topics and 10 prompts with 10 samples each, and compare against Best-of-N (Karthik et al., 2023) and FK-Steering (Singhal et al., 2025); SVDD-PM (Li et al., 2024a) collapses to degenerate text in this setting (GenPPL >1000). Full details are in Section G.

Figure 5 shows that DDNO scales more effectively than both search baselines. At small budgets all methods perform similarly, but Best-of-N and FK-Steering saturate once easy high-reward samples are exhausted, while DDNO keeps improving with every optimization step. At the largest budget, DDNO reaches a reward of 0.80 vs. 0.66 (Best-of-N) and 0.56 (FK-Steering), and this gain carries over to the held-out evaluator (0.38 vs. 0.22 and 0.19). Notably, the topic reward is indifferent to

fluency: DDNO's GenPPL is already elevated by the unoptimized surrogate (143 vs. 66) and rises only moderately during optimization (188), similar to FK-Steering, while Best-of-N stays close to the base model since it only selects among unmodified samples. This motivates a natural question: does the relationship change when the reward itself accounts for generation quality?

Open-ended quality reward. To answer this, we replace the topic classifier with a general-purpose quality reward model (QRM, `QRM-Llama3.1-8B` (Dorka, 2024)). QRM jointly scores helpfulness, truthfulness, and coherence, so higher reward now correlates with fluency. We evaluate on 10 open-ended explanatory prompts with 5 samples each and compare against compute-matched Best-of-N ($N{=}3T$; Section H). Table 3 shows that the reward–fluency relationship indeed reverses: DDNO raises QRM from 0.21 to 0.55 while *lowering* GenPPL from 40.2 to 11.6. At matched compute, it is on par with Best-of-N in reward, and at larger budgets becomes both more fluent and more consistent (Figure 10). Quality-aware rewards can thus improve fluency during optimization, whereas topic-only rewards do not prevent its degradation.

Table 3: **Quality reward (QRM)** at matched compute (T DDNO steps vs. $N{=}3T$ samples). Mean ± s.e.m. reported. Rewards are on par, while DDNO becomes more fluent and more consistent in PPL.

T / N	Method	QRM ↑	GenPPL ↓
–	Base	$0.21_{\pm.02}$	$40.2_{\pm14.7}$
10 / 30	Best-of-N	$0.46_{\pm.02}$	$14.4_{\pm1.5}$
	DDNO	$0.46_{\pm.02}$	$26.3_{\pm10.3}$
30 / 90	Best-of-N	$0.52_{\pm.01}$	$13.9_{\pm1.5}$
	DDNO	$0.52_{\pm.02}$	$13.0_{\pm2.4}$
50 / 150	Best-of-N	$0.54_{\pm.01}$	$14.4_{\pm1.5}$
	DDNO	$0.55_{\pm.02}$	$11.6_{\pm0.6}$

When does DDNO pay off? Our results suggest two regimes. Search-based methods can only select among samples the base model proposes, so they excel when high-reward outputs are common and saturate when they are rare, whereas DDNO actively moves toward such outputs. When high-reward outputs are rare, as in topic steering (base reward 0.05) and the compositional categories the base model rarely gets right, DDNO keeps improving where search plateaus. When they are common, as with QRM, where Best-of-N reaches most of its gains within 30 samples, search already finds good outputs quickly and DDNO is on par. DDNO is thus most valuable where search struggles: targets that are rare under the base model, at budgets large enough for gradients to pay off.

4 Related Work

Reward alignment of Discrete Diffusion models. For discrete diffusion, reward alignment has mostly been tackled at training time, via preference optimization or reinforcement learning (Borso et al., 2025; Zhao et al., 2025) or direct reward gradients, notably DRAKES (Wang et al., 2024), which also uses Gumbel-Softmax but updates model parameters rather than noise. At test time, existing discrete methods rely on guidance (Nisonoff et al., 2025; Schiff et al., 2025) or search (Li et al., 2024a; Singhal et al., 2025). DDNO brings test-time noise optimization, well established for continuous diffusion (Ben-Hamu et al., 2024; Eyring et al., 2024; Wallace et al., 2023), to the discrete setting. Gradient-based control of language models over hidden states or token logits (Dathathri et al., 2020; Kumar et al., 2021; Wang et al., 2026) operates in an unconstrained space prone to reward over-optimization (Gao et al., 2023); DDNO instead optimizes the initial noise, so every output remains reachable by the frozen model. We provide an extended discussion in Section D.

5 Discussion

We introduced DDNO, a test-time scaling framework that enables gradient-based optimization over the discrete initial noise in discrete diffusion models, via continuous noise logits and a differentiable surrogate based on GumbelSTE and soft mixing. DDNO keeps improving where search plateaus.

Limitations and outlook. DDNO requires differentiable rewards, and each iteration backpropagates through the full sampling chain, incurring 2–3× the cost of a forward pass plus memory overhead, with iterations run sequentially (Section E.3). The surrogate also introduces a fidelity gap (Table 1). Our text experiments further reveal that the reward–fluency trade-off is governed by the choice of reward function: narrow rewards such as topic classifiers can degrade fluency, while quality-aware rewards improve both. Few-step distilled text diffusion models (Lee et al., 2026; Potaptchik et al., 2026; Roos et al., 2026; Sahoo et al., 2025) could substantially reduce compute and memory costs.

AI USE STATEMENT

We used large language models as writing and coding assistants: to edit and polish the text, to help format tables and figures (including plotting scripts), and to check the manuscript for consistency. All research ideas, methods, experiments, and results are the authors' own. All reported numbers come from our experiments, and the authors reviewed and verified all LLM-assisted content and take full responsibility for the paper.

REFERENCES

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

Heli Ben-Hamu, Omri Puny, Itai Gat, Brian Karrer, Uriel Singer, and Yaron Lipman. D-flow: Differentiating through flows for controlled generation. In *ICML*, 2024.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation, 2013. URL https://arxiv.org/abs/1308.3432.

Julius Berner, Lorenz Richter, and Karen Ullrich. An optimal control perspective on diffusion-based generative modeling. *arXiv preprint arXiv:2211.01364*, 2022.

Ajinkya Bhole, Mohammad Mahmoudi Filabadi, Guillaume Crevecoeur, and Tom Lefebvre. Unifying entropy regularization in optimal control: From and back to classical objectives via iterated soft policies and path integral solutions. *arXiv preprint arXiv:2512.06109*, 2025.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *ICLR*, 2024.

Umberto Borso, Davide Paglieri, Jude Wells, and Tim Rocktäschel. Preference-based alignment of discrete diffusion models, 2025. URL https://arxiv.org/abs/2503.08295.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. *arXiv preprint arXiv:2402.04997*, 2024.

Arran Carter, Sanghyeok Choi, Kirill Tamogashev, Víctor Elvira, and Nikolay Malkin. Discrete diffusion samplers and bridges: Off-policy algorithms and applications in latent spaces. *arXiv preprint arXiv:2602.05961*, 2026.

George Casella and Christian P. Robert. Rao-blackwellisation of sampling schemes. *Biometrika*, 83 (1):81–94, 03 1996. ISSN 0006-3444. doi: 10.1093/biomet/83.1.81. URL https://doi.org/10.1093/biomet/83.1.81.

Chaofeng Chen, Annan Wang, Haoning Wu, Liang Liao, Wenxiu Sun, Qiong Yan, and Weisi Lin. Enhancing diffusion models with text-encoder reinforcement learning. In *ECCV*, 2024.

Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards. *arXiv preprint arXiv:2309.17400*, 2023.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1edEyBKDS.

Fei Deng, Qifei Wang, Wei Wei, Matthias Grundmann, and Tingbo Hou. Prdp: Proximal reward difference prediction for large-scale reward finetuning of diffusion models. In *CVPR*, 2024.

DiffusionGemmaTeam, Adrien Ali Taïga, James Assiene, Daniele Calandriello, Rahma Chaabouni, João Gante, Tamara von Glehn, Nate Keating, Chris Knutsen, Martin Kukla, Tianlin Liu, Ivan Lobov, Ofir Nabati, João Gabriel Oliveira, Nicolas Perez-Nieves, Nastasia Prutianova, Bobak Shahriari, Jean Tarbouriech, Pavel Tyletski, Çağlar Ünlü, Cindy Wu, Glenn Cameron, Jerome Connor, Sertan Girgin, Maarten Grootendorst, Alon Levkovitch, Eliya Nachmani, Omar Sanseviero, Piotr Stanczyk, Quentin Berthet, Andrew Campbell, Clément Crepy, Valentin De Bortoli, Arnaud Doucet, Romuald Elie, Alexandre Galashov, Klaus Greff, Alexis Jacq, David Ruhe, Yu-Han Wu, Sebastian Flennerhag, Brendan O'Donoghue, George Scrivener, and Shantanu Thakoor. Diffusiongemma technical report, 2026. URL https://arxiv.org/abs/2608.00146.

Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky TQ Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. *arXiv preprint arXiv:2409.08861*, 2024.

Nicolai Dorka. Quantile regression for distributional reward models in rlhf. *arXiv preprint arXiv:2409.10164*, 2024.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. *arXiv preprint arXiv:2403.03206*, 2024.

Luca Eyring, Shyamgopal Karthik, Karsten Roth, Alexey Dosovitskiy, and Zeynep Akata. Reno: Enhancing one-step text-to-image models through reward-based noise optimization. In *NeurIPS*, 2024.

Luca Eyring, Shyamgopal Karthik, Alexey Dosovitskiy, Nataniel Ruiz, and Zeynep Akata. Noise hypernetworks: Amortizing test-time compute in diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=DbzREoPwmM.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Reinforcement learning for fine-tuning text-to-image diffusion models. *NeurIPS*, 2023.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023.

Dhruba Ghosh, Hanna Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. In *NeurIPS*, 2023.

Xiefan Guo, Jinlin Liu, Miaomiao Cui, Jiankai Li, Hongyu Yang, and Di Huang. Initno: Boosting text-to-image diffusion models via initial noise optimization. In *CVPR*, 2024.

Mohsin Hasan, Viktor Ohanesian, Artem Gazizov, Yoshua Bengio, Alán Aspuru-Guzik, Roberto Bondesan, Marta Skreta, and Kirill Neklyudov. Discrete feynman-kac correctors. *arXiv preprint arXiv:2601.10403*, 2026.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*, 2021.

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

Peter Holderrieth, Douglas Chen, Luca Eyring, Ishin Shah, Giri Anantharaman, Yutong He, Zeynep Akata, Tommi Jaakkola, Nicholas Matthew Boffi, and Max Simchowitz. Diamond maps: Efficient reward alignment via stochastic flow maps. 2026. URL https://arxiv.org/abs/2602.05993.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Rohit Jena, Ali Taghibakhshi, Sahil Jain, Gerald Shen, Nima Tajbakhsh, and Arash Vahdat. Elucidating optimal reward-diversity tradeoffs in text-to-image diffusion models. *arXiv preprint arXiv:2409.06493*, 2024.

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 arXiv:2305.13308*, 2023.

Korrawe Karunratanakul, Konpat Preechakul, Emre Aksan, Thabo Beeler, Supasorn Suwajanakorn, and Siyu Tang. Optimizing diffusion noise can serve as universal motion priors. In *CVPR*, 2024.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Sachin Kumar, Eric Malmi, Aliaksei Severyn, and Yulia Tsvetkov. Controlled text generation as continuous optimization with multiple constraints. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=kTy7bbm-4I4.

Moritz Laurer, Wouter Van Atteveldt, Andreu Casas, and Kasper Welbers. Less annotating, more classifying: Addressing the data scarcity issue of supervised machine learning with deep transfer learning and bert-nli. *Political Analysis*, 32(1):84–100, 2024.

Chanhyuk Lee, Jaehoon Yoo, Manan Agarwal, Sheel Shah, Jerry Huang, Aditi Raghunathan, Seunghoon Hong, Nicholas M. Boffi, and Jinwoo Kim. One-step language modeling via continuous denoising, 2026. URL https://arxiv.org/abs/2602.16813.

Cheuk Kit Lee, Paul Jeha, Jes Frellsen, Pietro Lio, Michael Samuel Albergo, and Francisco Vargas. Debiasing guidance for discrete diffusion with sequential monte carlo. *arXiv preprint arXiv:2502.06079*, 2025.

Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. Aligning text-to-image models using human feedback. *arXiv preprint arXiv:2302.12192*, 2023.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gokcen Eraslan, Surag Nair, Tommaso Biancalani, Aviv Regev, Sergey Levine, and Masatoshi Uehara. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. *arXiv preprint arXiv:2408.08252*, 2024a.

Yanyu Li, Xian Liu, Anil Kag, Ju Hu, Yerlan Idelbayev, Dhritiman Sagar, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Textcraftor: Your text encoder can be image quality controller. In *CVPR*, 2024b.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *ICLR 2023*, 2023.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion language modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, et al. Inference-time scaling for diffusion models beyond scaling denoising steps. *arXiv preprint arXiv:2501.09732*, 2025.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

Pierre Marion, Anna Korba, Peter Bartlett, Mathieu Blondel, Valentin De Bortoli, Arnaud Doucet, Felipe Llinares-López, Courtney Paquette, and Quentin Berthet. Implicit diffusion: Efficient optimization through stochastic sampling, 2025. URL https://arxiv.org/abs/2402.05468.

Hunter Nisonoff, Junhao Xiong, Stephan Allenspach, and Jennifer Listgarten. Unlocking guidance for discrete state-space diffusion and flow models. *arXiv preprint arXiv:2406.01572*, 2024.

Hunter Nisonoff, Junhao Xiong, Stephan Allenspach, and Jennifer Listgarten. Unlocking guidance for discrete state-space diffusion and flow models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=XsgHl54yO7.

Zachary Novack, Julian McAuley, Taylor Berg-Kirkpatrick, and Nicholas J. Bryan. Ditto: Diffusion inference-time t-optimization for music generation, 2024. URL https://arxiv.org/abs/2401.12179.

Jingyang Ou, Jiaqi Han, Minkai Xu, Shaoxuan Xu, Jianwen Xie, Stefano Ermon, Yi Wu, and Chongxuan Li. Principled RL for diffusion LLMs emerges from a sequence-level perspective. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=S5YeC9llIL.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

Vincent Pauline, Tobias Höppe, Kirill Neklyudov, Alexander Tong, Stefan Bauer, and Andrea Dittadi. Foundations of diffusion models in general state spaces: A self-contained introduction. *arXiv preprint arXiv:2512.05092*, 2025.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

Peter Potaptchik, Jason Yim, Adhi Saravanan, Peter Holderrieth, Eric Vanden-Eijnden, and Michael S. Albergo. Discrete flow maps. 2026.

Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation. *arXiv preprint arXiv:2310.03739*, 2023.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Jarrid Rector-Brooks, Mohsin Hasan, Zhangzhi Peng, Cheng-Hao Liu, Sarthak Mittal, Nouha Dziri, Michael M. Bronstein, Pranam Chatterjee, Alexander Tong, and Joey Bose. Steering masked discrete diffusion models via discrete denoising posterior prediction. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Ombm8S40zN.

Christian P Robert and Gareth O Roberts. Rao-blackwellization in the mcmc era. *arXiv preprint arXiv:2101.01011*, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Daan Roos, Oscar Davis, Floor Eijkelboom, Michael Bronstein, Max Welling, İsmail İlkan Ceylan, Luca Ambrogioni, and Jan-Willem van de Meent. Categorical flow maps, 2026. URL https://arxiv.org/abs/2602.12233.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *arXiv preprint arXiv:2406.07524*, 2024.

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin T Chiu, and Volodymyr Kuleshov. The diffusion duality. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=9P9Y8FOSOk.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023.

Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dalla-torre, Bernardo P de Almeida, Alexander M Rush, Thomas PIERROT, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=i5MrJ6g5G1.

Neta Shaul, Itai Gat, Marton Havasi, Daniel Severo, Anuroop Sriram, Peter Holderrieth, Brian Karrer, Yaron Lipman, and Ricky T. Q. Chen. Flow matching with general discrete paths: A kinetic-optimal perspective, 2024. URL https://arxiv.org/abs/2412.03487.

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models, 2025. URL https://arxiv.org/abs/2501.06848.

Zhiwei Tang, Jiangweizhi Peng, Jiasheng Tang, Mingyi Hong, Fan Wang, and Tsung-Hui Chang. Inference-time alignment of diffusion models with direct noise optimization. *arXiv preprint arXiv:2405.18881*, 2024.

Masatoshi Uehara, Yulai Zhao, Kevin Black, Ehsan Hajiramezanali, Gabriele Scalia, Nathaniel Lee Diamant, Alex M Tseng, Tommaso Biancalani, and Sergey Levine. Fine-tuning of continuous-time diffusion models as entropy-regularized control, 2024. URL https://arxiv.org/abs/2402.15194.

Masatoshi Uehara, Xingyu Su, Yulai Zhao, Xiner Li, Aviv Regev, Shuiwang Ji, Sergey Levine, and Tommaso Biancalani. Reward-guided iterative refinement in diffusion models at test-time with applications to protein and dna design, 2025a. URL https://arxiv.org/abs/2502.14944.

Masatoshi Uehara, Yulai Zhao, Chenyu Wang, Xiner Li, Aviv Regev, Sergey Levine, and Tommaso Biancalani. Inference-time alignment in diffusion models with reward-guided generation: Tutorial and review, 2025b. URL https://arxiv.org/abs/2501.09685.

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. *arXiv preprint arXiv:2502.06999*, 2025.

Dimitri von Rütte, Janis Fluri, Omead Pooladzandi, Bernhard Schölkopf, Thomas Hofmann, and Antonio Orvieto. Scaling behavior of discrete diffusion language models. *arXiv preprint arXiv:2512.10858*, 2025.

Andrew Wagenmaker, Mitsuhiko Nakamoto, Yunchu Zhang, Seohong Park, Waleed Yagoub, Anusha Nagabandi, Abhishek Gupta, and Sergey Levine. Steering your diffusion policy with latent space reinforcement learning, 2025. URL https://arxiv.org/abs/2506.15799.

Bram Wallace, Akash Gokul, Stefano Ermon, and Nikhil Naik. End-to-end diffusion latent optimization improves classifier guidance. In *ICCV*, 2023.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *CVPR*, 2024.

Zhengyan Wan, Yidong Ouyang, Liyan Xie, Fang Fang, Hongyuan Zha, and Guang Cheng. Discrete guidance matching: Exact guidance for discrete flow matching. *arXiv preprint arXiv:2509.21912*, 2025.

Chenyu Wang, Masatoshi Uehara, Yichun He, Amy Wang, Tommaso Biancalani, Avantika Lal, Tommi Jaakkola, Sergey Levine, Hanchen Wang, and Aviv Regev. Fine-tuning discrete diffusion models via reward optimization with applications to dna and protein design. *arXiv preprint arXiv:2410.13643*, 2024.

Guanghan Wang, Yair Schiff, Gilad Turok, and Volodymyr Kuleshov. d2: Improved techniques for training reasoning diffusion language models, 2025a. URL https://arxiv.org/abs/2509.21474.

Jin Wang, Yao Lai, Aoxue Li, Shifeng Zhang, Jiacheng Sun, Ning Kang, Chengyue Wu, Zhenguo Li, and Ping Luo. Fudoki: Discrete flow-based unified understanding and generation via kinetic-optimal velocities. 2025b. URL https://arxiv.org/abs/2505.20147.

Peihao Wang, Ruisi Cai, Zhen Wang, Hongyuan Mei, Qiang Liu, Pan Li, and Atlas Wang. ∇-reasoner: Llm reasoning via test-time gradient descent in latent textual space. In *International Conference on Learning Representations*, 2026.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long papers)*, pp. 1112–1122, 2018.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025. URL https://arxiv.org/abs/2508.02324.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

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. In *NeurIPS*, 2023.

Oussama Zekri and Nicolas Boullé. Fine-tuning discrete diffusion models with policy gradient methods. *Advances in Neural Information Processing Systems*, 38, 2025.

Yinan Zhang, Eric Tzeng, Yilun Du, and Dmitry Kislyuk. Large-scale reinforcement learning for diffusion models. In *ECCV*, 2024.

Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. d1: Scaling reasoning in diffusion large language models via reinforcement learning, 2025. URL https://arxiv.org/abs/2504.12216.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

APPENDIX

A THEORETICAL FOUNDATIONS

A.1 NOTATION AND PRELIMINARIES

We work with discrete sequences $\mathbf{x} = (x^1, \dots, x^D) \in \mathcal{S} = \mathcal{T}^D$ over a finite alphabet $\mathcal{T} = \{1, \dots, K\}$. Path measures are denoted with calligraphic letters: $\mathcal{Q}$ for the generative prior, $\mathcal{P}$ for the tilted measure. Marginal distributions at time t are denoted $q_t(\mathbf{x})$ or $p_t(\mathbf{x})$. The transition kernel from time t to $s > t$ is $q(\mathbf{x}_s|\mathbf{x}_t)$.

For a path measure $\mathcal{P}$ on trajectories $\mathbf{x}_{[0,1]} = (\mathbf{x}_t)_{t\in[0,1]}$, we write $\mathcal{P}_{[0,1]}$ for the full path measure and $\mathcal{P}_t$ for the marginal at time t. The Radon-Nikodym derivative between absolutely continuous measures is denoted $\frac{d\mathcal{P}}{d\mathcal{Q}}$.

A.2 EXTENDED BACKGROUND

Our goal is to model a data distribution p_{data} over discrete sequences $\mathbf{x} \in \mathcal{S} = \mathcal{T}^D$, where $\mathcal{T} = \{1, \dots, K\}$ is the token alphabet and D is the sequence length. Discrete diffusion and flow matching both construct a *continuous-time path of marginals* that interpolates between a tractable source distribution $p_{\text{noise}}(\mathbf{x})$ and the data distribution $p_{\text{data}}(\mathbf{x})$. Diffusion emphasizes the forward noising process and its time reversal, while flow matching emphasizes the *velocity field* (rate matrix) whose Kolmogorov forward equation reproduces the chosen path.

Probability Paths. A probability path $\{q_t(\mathbf{x})\}_{t\in[0,1]}$ is a time-parameterized family of probability mass functions interpolating between source and target:

$$q_0(\mathbf{x}) = p_{\text{noise}}(\mathbf{x}), \qquad q_1(\mathbf{x}) = p_{\text{data}}(\mathbf{x}). \tag{14}$$

To ensure tractability, paths are typically constructed by conditioning on clean data points $\mathbf{x}_1 \sim p_{\text{data}}$ and employing a mean-field factorization:

$$q_t(\mathbf{x}|\mathbf{x}_1) = \prod_{i=1}^{D} q_t(x^i|x_1^i). \tag{15}$$

A canonical choice used in the FM literature is a *convex interpolation* between noise and data with scheduler $\alpha_t \in [0, 1]$:

$$q_t(x^i|x_1^i) = (1 - \alpha_t)p_{\text{noise}}(x^i) + \alpha_t\delta_{x_1^i}(x^i), \quad \alpha_0 = 0, \alpha_1 = 1, \tag{16}$$

where p_{noise} denotes the reference noise distribution. In this work we use a uniform noise distribution over the modeled vocabulary, $p_{\text{noise}}(x) = 1/K$, so the interpolation reduces to $q_t(x^i \mid x_1^i) = \alpha_t\delta_{x_1^i}(x^i) + (1 - \alpha_t)/K$.

Velocity Fields and Markov Dynamics. The path $\{q_t\}$ is realized through a continuous-time Markov chain (CTMC) $\{X_t\}_{t\in[0,1]}$ with marginals $X_t \sim q_t$. The dynamics are governed by a time-dependent rate matrix/velocity field $u_t(\mathbf{x}, \mathbf{z})$ satisfying Campbell et al. (2024); Lipman et al. (2023):

$$q_{t+h|t}(\mathbf{x}|\mathbf{z}) = \delta_{\mathbf{z}}(\mathbf{x}) + h\, u_t(\mathbf{x}, \mathbf{z}) + o(h), \tag{17}$$

subject to the standard CTMC constraints: $u_t(\mathbf{x}, \mathbf{z}) \geq 0$ for $\mathbf{x} \neq \mathbf{z}$ and $\sum_{\mathbf{x}} u_t(\mathbf{x}, \mathbf{z}) = 0$. Equivalently, for $x \neq y$ Pauline et al. (2025),

$$u_t(\mathbf{x}, \mathbf{y}) = \frac{\partial}{\partial s}\, q_{s|t}(\mathbf{y} \mid \mathbf{x})\Big|_{s=t^+}, \qquad \mathbf{x} \neq \mathbf{y}. \tag{18}$$

For computational tractability in high-dimensional sequences, updates are restricted to single-coordinate transitions:

$$u_t(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{D} u_t^i(x^i, \mathbf{z}) \prod_{j\neq i} \delta_{z^j}(x^j), \tag{19}$$

enabling position-wise independent sampling with $o(h)$ approximation error. The marginal velocity decomposes as

$$u_t^i(x^i, \mathbf{z}) = \sum_{x_1^i \in \mathcal{T}} u_t^i(x^i, z^i | x_1^i)\, q_{1|t}^i(x_1^i | \mathbf{z}), \tag{20}$$

where $q_{1|t}^i(x_1^i|\mathbf{z})$ represents the posterior probability of the clean token at position i. This posterior-averaging form is the discrete flow-matching identity: marginal velocity equals the posterior expectation of conditional velocities along the interpolant.

Learning the Posterior (clean-data estimation). The learnable component is a neural network f_θ that approximates the posterior over clean sequences given noised states. The network maps embedded sequences $U(\mathbf{x}_t)$ and time t to *logits* $\hat{\boldsymbol{\ell}}_{1|t} = f_\theta(U(\mathbf{x}_t), t)$. Applying a softmax yields a *soft clean-data estimate*: for each position i, the probability vector over clean tokens is

$$\hat{q}_{1|t}(x_1^i = k \mid \mathbf{x}_t, t) = \mathrm{softmax}(\hat{\boldsymbol{\ell}}_{1|t}^i)_k, \tag{21}$$

which is exactly the model's estimate of the posterior probability that token k is the clean token at position i given the noised sequence $\mathbf{x}_t$.

Sampling. Generation proceeds by simulating the *generative* CTMC from $t = 0$ (noise) to $t = 1$ (data), i.e., the time-reversal of the noising path. At each step $t \to s = t + h$ and each position i, the model prediction $\hat{q}_{1|t}^\theta$ is combined with the rate-matrix structure to compute a transition distribution $q_{s|t}^i(\cdot \mid \mathbf{x}_t)$, Equations (17), (19) and (20). A convenient equivalent implementation samples via a *jump/stay* decomposition: let $c = \bar{x}_t^i$ be the current token,

$$p_{\mathrm{jump}}^i \;\coloneqq\; 1 - q_{s|t}^i(c \mid \mathbf{x}_t), \tag{22}$$

$$r_{s|t}^i(k) \;\coloneqq\; \frac{q_{s|t}^i(k \mid \mathbf{x}_t)\, \mathbf{1}\{k \neq c\}}{p_{\mathrm{jump}}^i} \quad (\text{if } p_{\mathrm{jump}}^i > 0), \tag{23}$$

then sample a destination $y^i \sim \mathrm{Cat}(r_{s|t}^i)$ and a mask $m^i \sim \mathrm{Bernoulli}(p_{\mathrm{jump}}^i)$, and set $\bar{x}_s^i = c$ if $m^i = 0$ and $\bar{x}_s^i = y^i$ if $m^i = 1$. This two-stage update is exactly equivalent to sampling $\bar{x}_s^i \sim \mathrm{Cat}(q_{s|t}^i)$ (see Lemma A.1). Across positions, these categorical choices define a (non-differentiable) generative map $G_\theta : \mathcal{S} \to \mathcal{S}$. Consequently, gradients cannot flow from sequence-level objectives $R(\mathbf{x}_1)$ back through the sampling process, creating a barrier to direct gradient-based optimization.

A.3 Sampling via jump/stay decomposition

Lemma A.1 (Jump decomposition equals the categorical transition)**.** *Let $q \in \Delta^{K-1}$ and $c \in \{1, \dots, K\}$. Define $p_{\mathrm{jump}} = 1 - q(c)$ and $r(k) = \frac{q(k)\mathbf{1}\{k \neq c\}}{p_{\mathrm{jump}}}$ for $p_{\mathrm{jump}} > 0$. Let $m \sim \mathrm{Bernoulli}(p_{\mathrm{jump}})$ and $y \sim \mathrm{Cat}(r)$ independent, and set $x = c$ if $m = 0$ and $x = y$ if $m = 1$. Then $x \sim \mathrm{Cat}(q)$.*

Proof. Under $p_{\mathrm{jump}} > 0$: for $k = c$, $\mathbb{P}(x = c) = \mathbb{P}(m = 0) = 1 - p_{\mathrm{jump}} = q(c)$. For $k \neq c$, $\mathbb{P}(x = k) = \mathbb{P}(m = 1)\mathbb{P}(y = k) = p_{\mathrm{jump}} r(k) = q(k)$. □

A.4 Reward Tilting, Guidance, and Distributional Objective

The per-sample objective equation 6 used in the main text can be derived from a distributional reward-tilting perspective. We present this connection here for completeness.

Reward-tilted distribution. Given a base generative distribution $q(\mathbf{x})$ and reward $R : \mathcal{S} \to \mathbb{R}$, the *reward-tilted distribution* is

$$q_\beta(\mathbf{x}) \;=\; \frac{q(\mathbf{x})\, e^{\beta R(\mathbf{x})}}{Z_\beta}, \qquad Z_\beta \;=\; \mathbb{E}_{\mathbf{x} \sim q}\big[e^{\beta R(\mathbf{x})}\big], \tag{24}$$

where $\beta > 0$ controls alignment strength and Z_β is the intractable partition function.

Variational characterization. Let $\mathcal{Q}$ denote the path measure induced by the generative CTMC with terminal marginal $\mathcal{Q}_1 = p_{\text{data}}$. The tilted distribution is the terminal marginal of (Uehara et al., 2025a)

$$\mathcal{P}^\star = \arg\min_{\mathcal{P}} \Big\{ D_{\mathrm{KL}}(\mathcal{P}\|\mathcal{Q}) - \beta\, \mathbb{E}_{\mathcal{P}_1}[R(\mathbf{x}_1)] \Big\}, \qquad q_\beta = \mathcal{P}_1^\star. \tag{25}$$

Sampling from q_β is equivalent to solving a KL-regularized optimal control problem (Bhole et al., 2025; Levine, 2018).

Guidance and its limitations. The exact CTMC rates generating these tilted marginals require intractable posterior expectations at every step (Lee et al., 2025; Uehara et al., 2025b):

$$u_{t,\beta}(\mathbf{x}, \mathbf{z}) = u_t(\mathbf{x}, \mathbf{z})\, \frac{\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x})}[e^{\beta R(\mathbf{x}_1)}]}{\mathbb{E}_{q(\mathbf{x}_1|\mathbf{z})}[e^{\beta R(\mathbf{x}_1)}]}, \qquad \mathbf{x} \neq \mathbf{z}, \tag{26}$$

where $u_t(\mathbf{x}, \mathbf{z})$ is the CTMC transition rate from state $\mathbf{x}$ to $\mathbf{z}$ (Section A.2). Practical guidance methods (Nisonoff et al., 2025; Schiff et al., 2025) substitute plug-in approximations, replacing $\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_t)}[e^{\beta R(\mathbf{x}_1)}]$ with $e^{\beta R(\hat{\mathbf{x}}_1^t)}$. While efficient, this breaks the formal connection to q_β and degrades sample quality when posteriors are multimodal or rewards are non-smooth.

Distributional DDNO objective. To motivate DDNO, consider an idealized optimization problem in which only the initial distribution changes and the pretrained transition kernels remain fixed. Under this restriction, the path-space KL reduces to a divergence on initial conditions alone:

$$q_0^\star = \arg\max_{q_0} \Big\{ \mathbb{E}_{\mathbf{x}_0 \sim q_0}\big[V(\mathbf{x}_0)\big] - \tfrac{1}{\beta}\, \mathrm{KL}(q_0 \| p_{\text{noise}}) \Big\}, \tag{27}$$

where $V(\mathbf{x}_0) \coloneqq \mathbb{E}_{\mathcal{Q}}[R(\mathbf{x}_1) \mid \mathbf{x}_0]$ is the expected terminal reward under the frozen reverse process. Restricting to point masses $q_0 = \delta_{\mathbf{x}_0}$ and using the uniform prior recovers the per-sample objective equation 6 (Lemma A.4, Corollary A.5). This derivation characterizes the idealized unchanged-transition objective. The implemented DDNO procedure instead optimizes a fixed-randomness surrogate; in the ancestral setting, soft mixing modifies the forward sampling dynamics

A.5 Tilted Distribution and Variational Characterization

The tilted path measure reweights trajectories by their terminal reward only.

Proposition A.2 (ELBO Decomposition)**.** *For any path measure $\mathcal{P}$ with $\mathcal{P} \ll \mathcal{Q}$:*

$$D_{KL}(\mathcal{P}\|\mathcal{P}_\beta) = D_{KL}(\mathcal{P}\|\mathcal{Q}) - \beta\, \mathbb{E}_{\mathcal{P}}[R(\mathbf{x}_1)] + \log Z_\beta \tag{28}$$

Proof. By definition and the chain rule for Radon-Nikodym derivatives:

$$D_{KL}(\mathcal{P}\|\mathcal{P}_\beta) = \mathbb{E}_{\mathcal{P}}\left[\log \frac{d\mathcal{P}}{d\mathcal{P}_\beta}\right] = \mathbb{E}_{\mathcal{P}}\left[\log \frac{d\mathcal{P}}{d\mathcal{Q}} \cdot \frac{d\mathcal{Q}}{d\mathcal{P}_\beta}\right] \tag{29}$$

$$= \mathbb{E}_{\mathcal{P}}\left[\log \frac{d\mathcal{P}}{d\mathcal{Q}}\right] + \mathbb{E}_{\mathcal{P}}\left[\log \frac{Z_\beta}{e^{\beta R(\mathbf{x}_1)}}\right] \tag{30}$$

$$= D_{KL}(\mathcal{P}\|\mathcal{Q}) - \beta\, \mathbb{E}_{\mathcal{P}}[R(\mathbf{x}_1)] + \log Z_\beta. \tag{31}$$

□

Corollary A.3 (Variational Characterization)**.** *Since $D_{KL}(\mathcal{P}\|\mathcal{P}_\beta) \geq 0$ with equality iff $\mathcal{P} = \mathcal{P}_\beta$:*

$$\mathcal{P}_\beta = \arg\min_{\mathcal{P}} \left\{ D_{KL}(\mathcal{P}\|\mathcal{Q}) - \beta\, \mathbb{E}_{\mathcal{P}}[R(\mathbf{x}_1)] \right\}. \tag{32}$$

A.6 Stochastic Optimal Control Perspective

The path-space variational problem equation 27 is an instance of *KL-regularized stochastic optimal control* (Berner et al., 2022; Bhole et al., 2025; Levine, 2018). In this framework, an agent seeks to maximize expected reward while paying a cost for deviating from reference dynamics:

$$\max_{\mathcal{P}} \left\{ \mathbb{E}_{\mathcal{P}}[R(\mathbf{x}_1)] - \frac{1}{\beta}\, \mathrm{KL}(\mathcal{P}\|\mathcal{Q}) \right\}. \tag{33}$$

The KL term acts as a *control cost*: modifying the generative process to increase reward incurs a penalty proportional to the divergence from the pre-trained model $\mathcal{Q}$. The parameter β governs the exploration-exploitation trade-off—small β keeps generations close to the prior, while large β aggressively pursues reward.

We refer to the existing literature for an extensive study of this formulation in discrete or continuous spaces Bhole et al. (2025); Levine (2018)

A.7 Continuous Test-Time Noise Optimization.

Test-time noise optimization Ben-Hamu et al. (2024); Eyring et al. (2024); Guo et al. (2024); Karunratanakul et al. (2024); Novack et al. (2024); Tang et al. (2024); Wallace et al. (2023) techniques aim to improve or control pre-trained generative models on a per-sample basis at inference. Given a pre-trained generator g_θ, this approach optimizes the initial continuous noise $\mathbf{x}_0$ for each generation instance. The objective is to find an improved $\mathbf{x}_0^\star$ that maximizes a given reward $r(g_\theta(\mathbf{x}_0))$, often subject to regularization and can be formulated as

$$\mathbf{x}_0^\star = \arg\max_{\mathbf{x}_0}(r(g_\theta(\mathbf{x}_0)) - \mathrm{Reg}(\mathbf{x}_0)), \tag{34}$$

where $\mathrm{Reg}(\mathbf{x}_0)$ is a regularization term designed to keep $\mathbf{x}_0^\star$ within a high-density region of the prior, thus ensuring the generated sample $g_\theta(\mathbf{x}_0^\star)$ remains plausible.

A.8 Distribution-level vs. Per-sample DDNO

Lemma A.4 (Dirac restriction yields MAP objective)**.** *Restricting Equation* (27) *to Dirac measures* $q_0 = \delta_{\mathbf{x}_0}$ *yields*

$$\max_{\mathbf{x}_0 \in \mathcal{S}} \left\{ V(\mathbf{x}_0) + \tfrac{1}{\beta} \log p_{\textit{noise}}(\mathbf{x}_0) \right\}.$$

Proof. For $q_0 = \delta_{\mathbf{x}_0}$, we have $\mathbb{E}_{q_0}[V] = V(\mathbf{x}_0)$ and $\mathrm{KL}(\delta_{\mathbf{x}_0} \| p_{\text{noise}}) = -\log p_{\text{noise}}(\mathbf{x}_0)$. □

Corollary A.5 (Uniform prior eliminates Dirac regularization)**.** *When* $p_{\textit{noise}}$ *is uniform over* $\mathcal{S}$*, i.e.,* $p_{\textit{noise}}(\mathbf{x}) = K^{-D}$ *for all* $\mathbf{x} \in \mathcal{S}$*, the prior term* $\log p_{\textit{noise}}(\mathbf{x}_0) = -D \log K$ *is constant across all sequences. The objective then reduces to*

$$\max_{\mathbf{x}_0 \in \mathcal{S}} V(\mathbf{x}_0).$$

Under uniform noise and Dirac restriction, the variational framework imposes no regularization—every initial sequence is equally in-distribution.

Remark A.6 (Categorical relaxation restores regularization). Gradient-based optimization requires a continuous relaxation. While the forward pass decodes via $\arg\max$ (effectively a Dirac), the STE backward pass computes gradients through $\mathrm{softmax}(\boldsymbol{\ell}_0)$, implicitly defining a factorized categorical $q_{0,\boldsymbol{\ell}_0}$. The KL regularizer $\mathrm{KL}(q_{0,\boldsymbol{\ell}_0} \| p_{\text{noise}})$ acts on this implicit distribution: it is non-constant, connects to the variational objective equation 27, and prevents logit saturation that would otherwise degrade gradient quality.

B Proofs

B.1 Rao–Blackwellization of the mask-gated update

Proposition B.1 (Mask Rao–Blackwellization)**.** *Fix a position* i *and consider one scalar component of the one-hot state update. Let* $\mathcal{G}_0$ *contain the current state and jump probability, and let* $\mathcal{G}_1 = \mathcal{G}_0 \vee \sigma(Y)$ *additionally contain the sampled destination. For the Bernoulli-gated update*

$$X_s^i = (1 - M^i)X_t^i + M^i Y^i, \qquad M^i \sim \mathrm{Bernoulli}(p_{\mathrm{jump}}^i),$$

with M^i *conditionally independent of* Y^i *given* $\mathcal{G}_0$*, the Rao–Blackwellized update*

$$\widetilde{X}_s^i := \mathbb{E}[X_s^i \mid \mathcal{G}_1] = (1 - p_{\mathrm{jump}}^i)X_t^i + p_{\mathrm{jump}}^i Y^i$$

preserves the conditional mean and reduces conditional variance:

$$\mathbb{E}[\widetilde{X}_s^i \mid \mathcal{G}_0] = \mathbb{E}[X_s^i \mid \mathcal{G}_0], \qquad \mathrm{Var}(\widetilde{X}_s^i \mid \mathcal{G}_0) \le \mathrm{Var}(X_s^i \mid \mathcal{G}_0). \tag{35}$$

For vector-valued one-hot states, the same statement holds componentwise, and equivalently for the trace of the conditional covariance.

Proof. Mean preservation follows from the tower property:

$$\mathbb{E}[\widetilde{X}_s^i \mid \mathcal{G}_0] = \mathbb{E}\big[\mathbb{E}[X_s^i \mid \mathcal{G}_1] \mid \mathcal{G}_0\big] = \mathbb{E}[X_s^i \mid \mathcal{G}_0].$$

For the variance claim, apply the conditional law of total variance:

$$\mathrm{Var}(X_s^i \mid \mathcal{G}_0) = \mathbb{E}\big[\mathrm{Var}(X_s^i \mid \mathcal{G}_1) \mid \mathcal{G}_0\big] + \mathrm{Var}\big(\mathbb{E}[X_s^i \mid \mathcal{G}_1] \mid \mathcal{G}_0\big).$$

Since the first term is nonnegative,

$$\mathrm{Var}\big(\mathbb{E}[X_s^i \mid \mathcal{G}_1] \mid \mathcal{G}_0\big) \le \mathrm{Var}(X_s^i \mid \mathcal{G}_0).$$

Using $\widetilde{X}_s^i = \mathbb{E}[X_s^i \mid \mathcal{G}_1]$ gives Equation (35). Equality holds if and only if $\mathrm{Var}(X_s^i \mid \mathcal{G}_1) = 0$ almost surely, e.g., when the mask has no residual randomness. □

Remark. Proposition B.1 is a variance reduction statement for the random *state estimator* induced by the Bernoulli mask under the base sampler. It does not by itself imply a variance reduction guarantee for the DDNO *gradient estimator*, since $R(\cdot)$ is generally nonlinear and DDNO uses biased straight-through surrogate gradients.

C Algorithms

C.1 Surrogate with Adaptive Sampling

Algorithm 2 DDNO with Adaptive Sampling: ***Base Sampler*** vs. ***Deterministic Differentiable Surrogate***

Require: Model f_θ, steps $\frac{1}{h}$, length D, vocab K, updates-per-step m, sampling temperature T (base), relaxation temperature τ (surrogate), optional filtering (top-k/top-p), noise mask $\mathbf{n} \in \{0,1\}^D$, learnable noise logits $\boldsymbol{\ell}_0 \in \mathbb{R}^{D\times K}$, fixed per-step Gumbel noises $\xi = \{\boldsymbol{\epsilon}_t\}_t$.

Initialize at maximal noise ($t = 0$)**:**

Standard Prior	***Differentiable Init***
$\mathbf{x}_0 \sim \mathrm{Uniform}([K])^{\otimes D}$	$\boldsymbol{\ell}_0 \sim \mathcal{N}(0, \mathbf{I}), \quad \mathbf{x}_0 \leftarrow \textsc{Ste}(\boldsymbol{\ell}_0, \tau)$

for $t = 0, h, 2h, \ldots, 1-h$ (let $s = t + h$) **do**

$\lambda_t \leftarrow \textsc{LogSNR}(t)$

Predict token distributions:

$\mathbf{L}_t \leftarrow f_\theta(\mathbf{x}_t, \lambda_t), \quad \mathbf{p}_t \leftarrow \mathrm{softmax}(\mathbf{L}_t)$	$\mathbf{L}_t \leftarrow f_\theta(\tilde{\mathbf{x}}_t, \lambda_t), \quad \hat{\mathbf{p}}_t \leftarrow \mathrm{softmax}(\mathbf{L}_t)$ // *soft embedding for gradient flow*

Adaptive position selection (shared):

$\Delta_t^i \leftarrow \big(\max_k \hat{p}_t^i(k) - \hat{p}_t^i(\bar{x}_t^i)\big) \cdot \textsc{PriorProb}(\bar{x}_t^i, \lambda_t) \cdot n^i$

$\mathcal{I}_t \leftarrow \mathrm{TopK}(\Delta_t, m)$ *// no gradients through selection*

Proposal tokens:

(Sample / Argmax)	***(ST / Gumbel-ST with fixed ξ)***
$\mathbf{x}_{\text{prop}} \sim \mathrm{Cat}(\hat{\mathbf{p}}'_t)$	$\mathbf{x}_{\text{prop}} \leftarrow \begin{cases} \textsc{Ste}(\mathbf{L}'_t, \tau), & T = 0, \\ \textsc{GumbelSte}_{\boldsymbol{\epsilon}_t}(\mathbf{L}'_t/T, \tau), & T > 0 \end{cases}$

Update selected positions (shared):

$\mathbf{U}_t^i \leftarrow \mathbf{1}[i \in \mathcal{I}_t]$

$\mathbf{x}_s \leftarrow \mathbf{x}_t + \mathbf{U}_t \odot (\mathbf{x}_{\text{prop}} - \mathbf{x}_t)$

end for

Return: $\mathbf{x}_1$ (base), $\tilde{\mathbf{x}}_1$ (surrogate: soft probabilities for DDNO objective)

C.2 DDNO

Algorithm 3 Discrete Diffusion Noise Optimization (DDNO)

Input: Pre-trained model f_θ, reward function R, learning rate η, iterations N
Initialize $\boldsymbol{\ell}_0 \in \mathbb{R}^{D\times K} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $i = 1, \ldots, N$ **do**
  $\mathbf{x}_1^{\text{ste}} \leftarrow \text{SurrogateProcess}(f_\theta, \boldsymbol{\ell}_0)$ *// Algorithm 1*
  $J(\boldsymbol{\ell}_0) \leftarrow R(\mathbf{x}_1^{\text{ste}}) - \text{KL}(p_{\boldsymbol{\ell}_0} \| p_{noise})$
  $\boldsymbol{\ell}_0 \leftarrow \boldsymbol{\ell}_0 + \eta \nabla_{\boldsymbol{\ell}_0} J(\boldsymbol{\ell}_0)$
  if $J(\boldsymbol{\ell}_0) >$ best reward so far **then**
    Update best reward $\mathbf{x}_1^{\text{best}} \leftarrow \mathbf{x}_1^{\text{ste}}$
  end if
end for
Return: $\mathbf{x}_1^{\text{best}}$

D Extended Related Work

Discrete diffusion and flow models. Discrete diffusion and flow matching define generative processes over finite state spaces through continuous-time Markov chains (Austin et al., 2021; Campbell et al., 2024; Lou et al., 2023; Shaul et al., 2024). Masked (absorbing-state) diffusion has long been the dominant formulation for language (Sahoo et al., 2024), but uniform-noise models now match or exceed it (Sahoo et al., 2025; Schiff et al., 2025; von Rütte et al., 2025) and are deployed at scale, e.g., the 26B-parameter DiffusionGemma (DiffusionGemmaTeam et al., 2026). Discrete flows also underlie unified multimodal models that generate text and images in a shared token space (Wang et al., 2025b). Few-step distilled discrete models (Lee et al., 2026; Potaptchik et al., 2026; Roos et al., 2026; Sahoo et al., 2025) are a promising route to reducing the cost of unrolling the sampler, as required by DDNO.

Reward fine-tuning. Reward alignment for discrete diffusion has mostly been tackled at training time by fine-tuning the model, through preference optimization (Borso et al., 2025), reinforcement learning (Marion et al., 2025; Ou et al., 2026; Rector-Brooks et al., 2025; Wang et al., 2025a; Zhao et al., 2025), or direct reward gradients. Most related to our surrogate, DRAKES (Wang et al., 2024) also employs Gumbel-Softmax to obtain a differentiable generation, but uses the resulting gradients to update model parameters rather than the initial noise. Continuous diffusion models have likewise been aligned via reinforcement learning (Black et al., 2024; Chen et al., 2024; Deng et al., 2024; Fan et al., 2023; Venkatraman et al., 2025; Zhang et al., 2024) and direct reward fine-tuning (Clark et al., 2023; Domingo-Enrich et al., 2024; Jena et al., 2024; Lee et al., 2023; Li et al., 2024b; Prabhudesai et al., 2023; Uehara et al., 2024; Xu et al., 2023). All of these approaches are expensive and produce specialized checkpoints, whereas DDNO is training-free and leaves the model unchanged.

Test-time alignment of continuous diffusion. At test time, continuous diffusion models are improved by finding better initial noise or refining intermediate states under a reward. Search-based approaches evaluate multiple candidates (Karthik et al., 2023; Ma et al., 2025; Uehara et al., 2025a;b), while optimization-based approaches refine the noise by gradient ascent through the sampler (Ben-Hamu et al., 2024; Eyring et al., 2024; Guo et al., 2024; Holderrieth et al., 2026; Karunratanakul et al., 2024; Novack et al., 2024; Tang et al., 2024; Wallace et al., 2023). These methods rely on a deterministic, differentiable map from noise to output (e.g., an ODE solver or a distilled one-step model), which does not exist for discrete diffusion. DDNO constructs such a map for the discrete setting. A closely related line of work trains separate noise hypernetworks or latent policies that output reward-aligned initial noise (Carter et al., 2026; Eyring et al., 2025; Venkatraman et al., 2025; Wagenmaker et al., 2025). These methods amortize per-sample noise optimization into a learned mapping. DDNO is complementary: it provides the per-instance optimizer in the discrete setting, whose solutions such a hypernetwork could distill.

Test-time alignment of discrete diffusion. Existing training-free methods for discrete diffusion either modify the transition rates through per-step guidance (Nisonoff et al., 2025; Schiff et al., 2025; Wan et al., 2025), which relies on approximations of intractable posterior expectations (Section A.4), or search over intermediate states with sequential Monte Carlo and value-based decoding (Li et al.,

2024a; Singhal et al., 2025). Search-based methods select among samples the base model proposes, guided by intermediate reward estimates; we show empirically that these estimates are weakly predictive of the final reward and that search saturates as compute grows (Sections 3.2 and E.3). DDNO instead optimizes the initial noise directly along the gradient of the terminal reward, and is, to our knowledge, the first noise-optimization framework for discrete diffusion models.

Gradient-based test-time optimization for text. A related line of work controls frozen language models by optimizing continuous quantities at inference time. Plug-and-Play Language Models (Dathathri et al., 2020) perturb hidden activations using gradients from attribute models, while MuCoCo (Kumar et al., 2021) formulates constrained text generation as continuous optimization over relaxed token representations. Most closely in spirit, ∇-Reasoner (Wang et al., 2026) performs test-time gradient descent over latent textual representations or token logits to improve reasoning under reward and likelihood objectives. DDNO differs in *what* is optimized. Directly optimizing token logits or free embeddings operates in an unconstrained space easily exploited by the reward signal, often leading to reward over-optimization (Gao et al., 2023). DDNO instead optimizes the initial noise of a discrete diffusion model, a more natural target: it passes through the frozen reverse process, which acts as a generative bottleneck that limits drift from natural language, and the uniform noise prior provides a principled KL regularizer at no extra cost.

E Additional Results

E.1 Initialization of the Noise Logits

Section 2.3 initializes the noise logits as $\boldsymbol{\ell}_0 \sim \mathcal{N}(0, \sigma^2 I)$ with σ=1 for images and σ=0.05 for text. Here we show that neither the distribution family nor σ changes the initial sample, identify the single quantity through which they affect optimization, and evaluate its effect empirically.

Distribution family. The initial state is $x_0^i = \arg\max_k \ell_0^{i,k}$. For any i.i.d. distribution with a continuous marginal, the K logits at each position are exchangeable and ties occur with probability zero, so x_0^i is exactly uniform over $\{1, \dots, K\}$ and matches the prior p_{noise}. Gaussian, Laplace, Logistic, and Gumbel logits are therefore all valid; the Gaussian is a convenience, not a requirement.

Scale. Since $\arg\max$ is invariant to positive rescaling, σ has no effect on x_0, on the forward pass, or on the generated trajectory. It enters only the backward pass, through the sharpness of $\mathrm{softmax}(\boldsymbol{\ell}_0/\tau)$ in the straight-through estimator, and only via the ratio σ/τ. The scale is therefore not an independent hyperparameter: it shares one degree of freedom with the temperature τ. A convenient invariant coordinate is the normalized initial entropy

$$\bar{H} = \frac{1}{D \log K} \sum_{i=1}^{D} H\big(\mathrm{softmax}(\boldsymbol{\ell}_0^i/\tau)\big) \in [0, 1], \tag{36}$$

which is also the quantity controlled by the entropy regularizer in Equation (10). As $\bar{H}$ decreases, softmax coordinates saturate, the softmax Jacobian vanishes, and the corresponding logits stop receiving gradient signal. To second order in σ/τ, Gaussian logits give

$$\bar{H} \approx 1 - \frac{(\sigma/\tau)^2}{2 \log K}, \tag{37}$$

so $\bar{H}$ can be computed before any run and depends on the vocabulary size only logarithmically. Table 4 compares our two settings: the 20$\times$ difference in raw σ corresponds to only a 5% difference in $\bar{H}$, and both lie in the near-uniform regime.

Table 4: **Initial logit scale across domains.** Both settings initialize close to uniform, despite a 20$\times$ difference in σ.

Setting	σ	τ	σ/τ	$\bar{H}$
Text (GIDD-Unif-3B)	0.05	0.85	0.059	0.9998
Image (FUDOKI)	1.0	1.0	1.00	0.9485

Empirical sweep. We vary only σ on FUDOKI (8 prompts, paired seeds, 50 optimization steps, all other hyperparameters as in Table 10). Figure 6 and Table 5 show that the reward improves as the initialization approaches uniform, tracking $\bar{H}$ throughout. The first three settings overlap within one standard deviation, but the degradation at σ=4 is clear: $\bar{H}$ collapses to 0.38, and the entropy regularizer recovers it only to 0.56 during optimization. The scale used for text, σ=0.05, performs best on images as well, so the two domain settings lie on a single curve rather than reflecting domain-specific tuning. We did not re-tune σ for the main image results, which use σ=1.

Table 5: **Sweep over the initial logit scale** (FUDOKI). Reward is reported as mean ± std over prompts.

σ	$\bar{H}$ at init.	Reward ↑	$\bar{H}$ after opt.
0.05	0.9999	-2.24 ± 0.29	0.959
0.25	0.9968	-2.30 ± 0.34	0.954
1.0	0.9485	-2.58 ± 0.36	0.907
4.0	0.3794	-2.71 ± 0.39	0.564

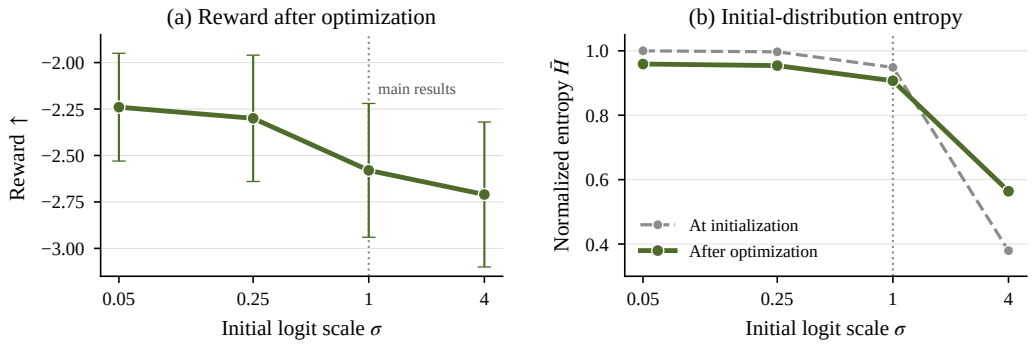

Figure 6: **Sensitivity to the initial logit scale** σ (FUDOKI, 8 prompts, 50 steps). σ does not change the initial sample, only gradient quality via the initial entropy $\bar{H}$. Reward improves as initialization approaches uniform; the first three settings overlap within one standard deviation, while σ=4 saturates the logits and the entropy regularizer cannot recover. Main results use σ=1.

Practical recipe. The choice is distribution-agnostic: initialize near-uniform, i.e., with $\bar{H}$ close to 1. Exactly uniform logits (σ=0) are degenerate, since the argmax then breaks ties deterministically (Section 2.3), so a small non-zero scale is needed; σ=0.05 is a safe default in both domains.

E.2 KL Regularization

We use a default KL penalty strength of $\lambda_{\mathrm{KL}} = 100$ and ablate this choice in Figure 7. Without regularization ($\lambda_{\mathrm{KL}} = 0$), the optimized distribution drifts substantially from the prior, as evidenced by the steadily increasing KL divergence in Figure 7(b). In contrast, $\lambda_{\mathrm{KL}} = 100$ effectively anchors the distribution close to the prior throughout optimization. Notably, this comes at no cost to performance: as shown in the zoomed inset of Figure 7(a), the KL-regularized variant achieves slightly higher best-so-far reward, suggesting that constraining the distribution helps avoid reward hacking and leads to more effective optimization. Table 6 confirms this observation where we find that leaving out the KL results in marginally worse performance on GenEval after 50 optimization steps.

KL Weight	GenEval Mean, 50 Opt. steps
0	0.91
100	**0.93**

Table 6: KL ablation in image generation.

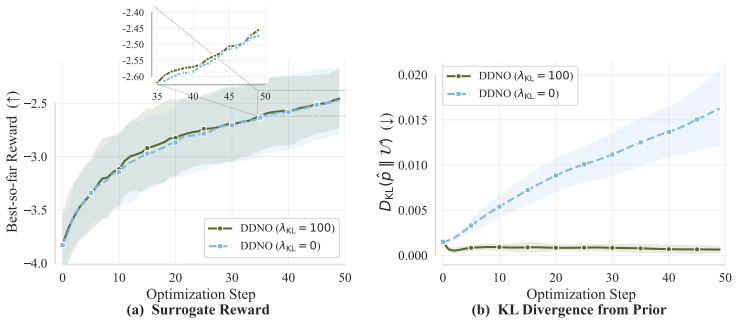

Figure 7: Ablation of KL regularization in DDNO. **(a)** Best-so-far surrogate reward over optimization steps. Both variants improve steadily; the zoomed inset reveals that $\lambda_{\mathrm{KL}} = 100$ attains marginally higher reward. **(b)** KL divergence from the prior. Without regularization ($\lambda_{\mathrm{KL}} = 0$), the distribution diverges increasingly from the prior, whereas $\lambda_{\mathrm{KL}} = 100$ keeps it tightly anchored.

E.3 Compute Matching and Cost

Protocol. All methods are timed on a single NVIDIA H100 (80 GB), and we match them in total GPU-hours. Each DDNO step consists of a forward and a backward pass through the unrolled surrogate, including the reward model, and is therefore more expensive than a forward generation. Under the standard convention that a forward pass costs $\approx 2PD$ FLOPs and a forward–backward pass $\approx 6PD$ FLOPs for a model with P parameters on D tokens, one step costs $\approx 3\times$ a forward generation. The measured wall-clock overhead is slightly lower (Table 7), so we conservatively charge each step $3\times$.

Images. We run DDNO with 16 NFE per surrogate pass, while all sampling-based methods use 50 NFE. One DDNO step ($\approx 3 \times 16 = 48$ forward-equivalent NFE) thus matches one Best-of-N sample, so T optimization steps correspond to $N{=}T$ samples, as on the two x-axes of Figure 3. SVDD-PM and FK-Steering are matched in GPU-hours through their population sizes M and P (Table 8).

Text. DDNO and all baselines use the same 128-step denoising loop, so T optimization steps are matched with $N{=}3T$ Best-of-N samples; FK-Steering uses $P \in \{25, 63, 125, 188\}$ particles for $T \in \{10, 25, 50, 75\}$.

Table 7: **Memory and wall-clock cost per iteration** on one H100. One DDNO step is compared with one Best-of-N sample (forward generation and reward evaluation).

Domain	Method	VRAM (GB)	Wall-clock / iteration
Text (GIDD-Unif-3B)	Best-of-N	16.7	1.0×
	DDNO	21.2	2.9×
Image (FUDOKI)	Best-of-N	26.9	1.0×
	DDNO	55.4	2.6×

Memory. DDNO requires $1.3\times$ (text) to $2.1\times$ (images) the memory of a forward generation, due to the activations stored for backpropagation; we use gradient checkpointing throughout. The overhead is larger for images because gradients are propagated through two large reward models (`HPSv2.1` and `NVILA-Lite-2B-Verifier`).

Search-based baselines. For FK-Steering (Singhal et al., 2025), we sweep the guidance strength $\lambda \in \{0.5, 1, 2, 5, 10, 20\}$ and report the best setting (λ=1 for FUDOKI, λ=10 for topic steering), resampling whenever the effective sample size drops below $P/2$. We return the highest-reward particle of the final population, i.e., Best-of-P on top of guided sampling, which is the most favorable evaluation protocol for this baseline. FK-Steering processes all P particles of a prompt in a single batched forward pass, but is more expensive per particle than Best-of-N because it evaluates the reward at every resampling step; it therefore affords fewer particles at matched compute. For SVDD-PM (Li et al., 2024a), we use the training-free variant with argmax decoding, as recommended by the authors. In topic steering, SVDD-PM collapsed to degenerate outputs (GenPPL > 1000) and is omitted from Figure 5.

Table 8: **Compute-matched results on GenEval (FUDOKI)** underlying Figure 3. Budgets are total GPU-hours. Bold marks the best method per budget. SVDD-PM and FK-Steering were not evaluated at $\sim$13 GPU-h.

Budget (GPU-h)	$\sim$6	$\sim$9.7	$\sim$13	$\sim$18–20	$\sim$25–26
Reward ↑					
Best-of-N (N)	-2.66 (16)	-2.63 (24)	-2.62 (32)	-2.60 (50)	-2.59 (64)
SVDD-PM (M)	$\mathbf{-2.63}$ (5)	$\mathbf{-2.57}$ (10)	–	-2.52 (20)	-2.51 (28)
FK-Steering (P)	-2.70 (5)	-2.63 (10)	–	-2.58 (20)	-2.56 (28)
DDNO (T)	-2.73 (16)	-2.59 (24)	$\mathbf{-2.50}$ (32)	$\mathbf{-2.34}$ (50)	$\mathbf{-2.25}$ (64)
GenEval (%) ↑					
Best-of-N (N)	85.32	87.93	**89.42**	90.39	91.35
SVDD-PM (M)	85.35	88.92	–	90.57	90.38
FK-Steering (P)	**86.48**	**89.58**	–	91.27	92.15
DDNO (T)	79.76	83.50	87.86	**92.21**	**93.83**

Sequential steps and parallelism. GPU-hours measure total compute, but the methods differ in how that compute can be parallelized. Per sample, Best-of-N requires S sequential denoiser calls, whereas DDNO requires $T \times S$ sequential calls plus the corresponding backward passes, since its optimization steps depend on each other. Which accounting is appropriate depends on what fills the batch dimension:

- *Throughput regime.* When many prompts are processed concurrently, as in batch serving, DDNO batches across prompts exactly as Best-of-N batches across candidates. Wall-clock time per prompt then tracks total compute per prompt, which is what our GPU-hour matching measures.
- *Latency regime.* For a single prompt, no batching compresses DDNO's serial depth below $T \times S$, whereas search-based methods can generate their candidates in parallel and reach a given budget faster.

In our timing, prompts are processed sequentially for all methods; FK-Steering batches its P particles within a prompt, while Best-of-N and DDNO are run without batching.

To quantify the latency regime, we benchmark batched Best-of-N on one H100 (N=64 candidates, one FUDOKI prompt). Increasing the generation batch size from 1 to 8, 16, and 32 yields wall-clock speedups of $1.65\times$, $1.97\times$, and $2.01\times$, with peak reserved memory rising to 30.5, 44.0, and 71.0 GiB. Batching thus helps, but saturates around $2\times$ as memory becomes the bottleneck on an 80 GB GPU; FK-Steering and SVDD-PM face analogous memory pressure as their populations grow. Applied to Figure 3, this would shift the GenEval crossover between DDNO and Best-of-N from roughly 15 to roughly 30 GPU-hours. This estimate lies at the edge of our measured range. DDNO is therefore best suited to settings where output quality matters more than single-prompt latency. Importantly, independent DDNO runs, e.g., from different seeds or different prompts, remain fully parallelizable.

F Image Generation Experiment Details

This section provides implementation details for the compositional text-to-image experiments in Section 3.2. All experiments are conducted on a single H100 GPU.

F.1 Model and Architecture

We use FUDOKI (Wang et al., 2025b), a discrete flow matching model with kinetic-optimal paths (Shaul et al., 2024). FUDOKI operates on discrete image tokens and generates images through a non-autoregressive denoising process with a uniform noise prior. The model uses a transformer backbone trained on large-scale image–text pairs.

F.2 Benchmark and Evaluation

GenEval Benchmark. We evaluate on GenEval (Ghosh et al., 2023), a benchmark designed to measure compositional understanding in text-to-image models. GenEval evaluates six categories of compositional capabilities:

- **Single Object**: generating a single specified object
- **Two Objects**: generating two distinct objects in the same scene
- **Counting**: generating the correct number of objects
- **Colors**: correctly applying specified colors to objects
- **Position**: placing objects in specified spatial relationships
- **Attribute Binding**: correctly associating attributes with their corresponding objects

Each category contains multiple prompts, and evaluation is performed using automated object detection and attribute verification. GenEval is never used during optimization and thus serves as a held-out measure of prompt alignment.

F.3 Reward Models

Our reward combines two components to balance semantic fidelity with aesthetic quality.

Semantic Fidelity Reward. We use NVILA-Lite-2B-Verifier, a vision-language model that scores the alignment between generated images and text prompts. Given an image–prompt pair, the model is queried with:

> *"You are an AI assistant specializing in image analysis and ranking. Your task is to analyze and compare an image based on how well it matches the given prompt. The given prompt is:* `{prompt}`*. Please consider the prompt and the image to make a decision and respond directly with 'yes' or 'no'."*

The reward is the probability assigned to the token `'yes'`, extracted from the model's output logits.

Aesthetic Quality Reward. We use HPSv2.1 (Human Preference Score v2.1) (Wu et al., 2023), a model trained on human preference data to predict the aesthetic quality of generated images.

Combined Reward. The final reward is the unweighted sum

$$R = R_{\text{fidelity}} + R_{\text{aesthetic}}. \tag{38}$$

The same reward is used for DDNO and all reward-based baselines.

F.4 Baseline Methods

We compare DDNO against the following inference-time baselines. For all methods except CFG, we set the CFG scale to 0 to cleanly disentangle the performance of each method.

Base Model (FUDOKI). Standard sampling from the pretrained FUDOKI model using the default sampling configuration (50 NFE), without any reward-based steering.

Classifier-Free Guidance (CFG). We apply classifier-free guidance (Ho & Salimans, 2022) with the guidance scale swept over $\{1.5, 2.0, 3.0, 5.0, 7.5, 10.0\}$ and report the best-performing configuration.

Table 9: **Ablation of the reward-guidance scale** on GenEval. All runs use CFG $= 0$ and 50 NFE.

Method	Guidance Scale	GenEval Mean
FUDOKI Base	0	0.656
+ Reward Guidance	50	0.713
+ Reward Guidance	75	**0.745**
+ Reward Guidance	100	0.695
+ Reward Guidance	125	0.635
+ Reward Guidance	150	0.620
+ Reward Guidance	200	0.526

Reward Guidance. We implement discrete reward guidance (Nisonoff et al., 2025) with the same reward as DDNO. We sweep the guidance strength (Table 9) and report the best configuration (λ=75). Reward guidance improves over the base model at moderate strength, but larger strengths quickly over-optimize the reward and degrade GenEval.

Best-of-N Sampling. For each prompt, we generate N independent samples from the base model (50 NFE) and select the one with the highest reward (Karthik et al., 2023; Ma et al., 2025). Best-of-N is matched to DDNO in total compute: since DDNO runs the surrogate at 16 NFE and each step costs $\approx$3$\times$ a forward pass, N samples correspond to T=N optimization steps (Section E.3). We report $N \in \{16, 24, 32, 50, 64\}$, with N=64 in Table 2.

SVDD-PM. We use the training-free variant of SVDD (Li et al., 2024a) with argmax decoding, as recommended by the authors, and the same reward. At each denoising step, SVDD-PM draws M candidate next states and selects one according to the reward of its one-step estimate of the final sample. We use $M \in \{5, 10, 20, 28\}$, matched to the other methods in GPU-hours (Section E.3).

FK-Steering. We use Feynman–Kac steering (Singhal et al., 2025) with P particles, the same reward as potential, and resampling whenever the effective sample size drops below $P/2$. We sweep the guidance strength $\lambda \in \{0.5, 1, 2, 5, 10, 20\}$ and use the best setting (λ=1). We return the highest-reward particle of the final population, i.e., Best-of-P on top of guided sampling, which is the most favorable evaluation protocol for this baseline. We use $P \in \{5, 10, 20, 28\}$, matched in GPU-hours (Section E.3). Since SVDD-PM and FK-Steering evaluate the reward at every denoising step rather than once per sample, a given budget affords fewer particles than Best-of-N samples (e.g., P=28 vs. N=64 at $\sim$26 GPU-h); with the cheaper text reward, the counts are closer (P=188 vs. N=225).

Differentiable Surrogate (Unoptimized). The differentiable surrogate of Section 2.2 without any optimization of the initial noise logits. This baseline quantifies the fidelity gap introduced by the surrogate.

F.5 DDNO Hyperparameters

Table 10 summarizes the hyperparameters used for DDNO on GenEval.

Denoising Configuration. All baselines use the default FUDOKI configuration with 50 denoising steps. During optimization, DDNO runs the surrogate with 16 denoising steps to reduce the cost per step; the optimized noise transfers across step counts (Figure 8).

Optimization Details. The initial noise logits are optimized with Adam (Kingma, 2014). We use gradient checkpointing to reduce memory consumption during backpropagation through the full denoising trajectory, and run all experiments in mixed precision (BF16).

Table 10: **DDNO hyperparameters for image generation.**

HYPERPARAMETER	VALUE
OPTIMIZATION STEPS T	64 (OR COMPUTE-MATCHED)
NFE PER SURROGATE PASS	16
LEARNING RATE	0.1
OPTIMIZER	ADAM
SURROGATE TEMPERATURE τ	1.0
INITIAL LOGIT SCALE σ	1.0
GRADIENT CLIP NORM	1.0
KL REGULARIZATION WEIGHT λ_{KL}	100.0

F.6 EVALUATION PROTOCOL

Sample Generation. For each GenEval prompt, we run DDNO from freshly initialized noise logits and return the highest-reward sample across all optimization steps (Algorithm 3).

GenEval Scoring. We use the official GenEval evaluation pipeline, which employs object detection models to verify the presence and attributes of the specified objects. Each prompt is scored as pass/fail based on whether all compositional requirements are satisfied.

F.7 LIMITATIONS AND FAILURE MODES

Surrogate Fidelity Gap. As shown in Table 1, the unoptimized surrogate incurs a small fidelity drop compared to standard discrete sampling (0.65 vs. 0.67 GenEval). DDNO optimization more than compensates for this gap, and optimized noise decoded with the original sampler still reaches 0.86 (Table 2).

Optimization Cost. Each DDNO step requires a forward and a backward pass through the unrolled sampler, and steps run sequentially. At small budgets, search-based methods are therefore preferable; DDNO pays off at larger budgets, where search plateaus, and when high-reward outputs are rare under the base model (Section E.3).

Reward Model Dependence. The quality of DDNO's outputs depends on the reward model. Imperfect reward models may lead to reward hacking, where outputs achieve high reward without satisfying the compositional requirements. We monitor this with the held-out GenEval score, which rises together with the optimized reward throughout (Figure 3), and the KL regularizer further mitigates it (Section E.2).

G TOPIC STEERING EXPERIMENT DETAILS

This section provides implementation details for the topic steering experiments in Section 3.3. All experiments are conducted on a single H100 GPU.

G.1 MODEL AND ARCHITECTURE

We use GIDD-Unif-3B (von Rütte et al., 2025), a 3B-parameter uniform-noise discrete diffusion language model. We generate sequences of 64 tokens (excluding the prompt prefix) using 128 denoising steps with a cosine noise schedule and adaptive sampling (Section 2.2; Algorithm 2 for the surrogate).

G.2 TASK CONFIGURATION

We steer generation toward two topics (science and sports), using natural language inference (NLI) as the reward: the entailment probability between the generated text and the hypothesis "This text is

about {topic}." We evaluate on 10 narrative prompts designed as neutral starting points that do not bias the model toward either topic (Table 11), yielding 20 prompt–topic pairs.

Table 11: **Topic steering prompts.** Each prompt is evaluated with both the science and the sports target.

#	Prompt
1	On a gray Monday morning, Martin badges into the office a little early, coffee in hand, still buzzing from last night.
2	The train pulled out as Leila reread the invitation on her phone, wondering what she'd just agreed to.
3	Outside the meeting room, a cart of sealed boxes sat in the hall with a note: Open when everyone arrives.
4	Jonas stopped at the edge of the empty field, heard a distant whistle, and unfolded a printout he'd kept to himself.
5	The email subject read "We need answers", and the attachment was a single image with a red circle around something small.
6	Nina set a small container on the table. The label was missing, but the date and initials were still there.
7	At 2:13 a.m., the group chat lit up with a blurry photo and one message: Tell me this is normal.
8	The auditorium lights dimmed and the first slide appeared—no title, just a curve that rose fast and then leveled off.
9	He checked his watch, then the sky, then the stopwatch again, like time was messing with him.
10	When the door opened, disinfectant and chalk dust hit the air, and Aisha saw the room had been rearranged overnight.

G.3 Reward and Evaluation Models

Reward Model (Optimization). We use DeBERTa-v3-Large (He et al., 2021) fine-tuned on MNLI, FEVER, ANLI, LingNLI, and WANLI (Laurer et al., 2024) (`MoritzLaurer/DeBERTa-v3-large-mnli-fever-anli-ling-wanli`). The reward is the softmax probability of the "entailment" class for the hypothesis "This text is about {topic}."

Evaluator Model (Held-out). We use RoBERTa-Large fine-tuned on MultiNLI (Liu et al., 2019) (`roberta-large-mnli`) as a held-out evaluator. It provides an independent assessment of topic adherence and is never used during optimization or selection.

Perplexity Model. We measure fluency with GPT-2 XL (Radford et al., 2019) perplexity (GenPPL), computed as the exponentiated cross-entropy over the generated tokens.

G.4 Baselines

All baselines use the same denoising configuration as DDNO (128 steps, cosine schedule, adaptive sampling at temperature 0.85) and are matched in total compute (Section E.3).

Best-of-N Sampling. For each prompt and topic, we generate N independent samples from the base model and select the one with the highest reward. Since all methods share the same 128-step loop, T DDNO steps are matched with $N{=}3T$ samples; we report $N \in \{30, 75, 150, 225\}$ and the base model ($N{=}1$).

FK-Steering. We use Feynman–Kac steering (Singhal et al., 2025) with $P \in \{25, 63, 125, 188\}$ particles, matched in GPU-hours to $T \in \{10, 25, 50, 75\}$, resampling whenever the effective sample size drops below $P/2$, and returning the highest-reward particle of the final population. We sweep $\lambda \in \{0.5, 1, 2, 5, 10, 20\}$ and use $\lambda{=}10$. Because most particles receive near-zero topic reward, FK-Steering almost never resamples for $\lambda \leq 2$ (mean ESS$/P \geq 0.998$) and then reduces to Best-of-P; resampling becomes active only at $\lambda{=}10$ (mean ESS$/P$ of 0.975). Since the NLI reward model is small, reward evaluations at every resampling step add little cost, so particle counts are close to the Best-of-N sample counts.

SVDD-PM. We use the training-free variant of SVDD (Li et al., 2024a) with argmax decoding. SVDD-PM collapses to degenerate outputs in this setting (GenPPL >1000) and is therefore omitted from Figure 5; we analyze this failure below.

G.5 Why SVDD-PM Fails on Topic Steering

SVDD-PM selects among M candidate next states by the reward of their one-step (Tweedie) estimate of the final sample. With the sparse topic reward, this estimate decouples from the true reward: as M grows, the estimate stays at 0.06–0.15 while the reward of the actual final samples collapses from 0.11 to ≈0 (Table 12). Additional search thus exploits the lookahead estimate rather than improving the output.

Table 12: **SVDD-PM on topic steering.** Reward of the final sample vs. reward of the Tweedie estimate used for selection.

M	Final-sample reward	Tweedie-estimate reward
5	0.112	0.133
10	0.003	0.059
20	0.001	0.151

G.6 DDNO Hyperparameters

Table 13 summarizes the DDNO hyperparameters used for topic steering.

Table 13: **DDNO hyperparameters for topic steering.** Used for all prompts and both topics; T=0 corresponds to the unoptimized surrogate.

Hyperparameter	Value
Optimization steps (T)	0 / 10 / 25 / 50 / 75
Learning rate	0.05
Guidance scale (β)	10 000
KL weight (λ_{KL})	0.1
Surrogate temperature (τ)	0.85
Initial logit scale (σ)	0.05
Gradient clip norm	0.001
Sequence length (suffix)	64
Denoising steps	128
Samples per prompt and topic	10

Optimization Details. We optimize the initial noise logits $\boldsymbol{\ell}_0 \in \mathbb{R}^{L\times V}$ (L=64, V the GIDD vocabulary size) with Adam (Kingma, 2014), initialized from $\mathcal{N}(0, 0.05^2 I)$. Each step backpropagates through the full 128-step denoising loop using GumbelSTE with fixed per-step Gumbel noise, with gradient checkpointing to reduce memory. We minimize

$$\mathcal{L} = -\beta \cdot R(\mathbf{x}) + \lambda_{\text{KL}} \cdot D_{\text{KL}}(\text{softmax}(\boldsymbol{\ell}_0) \,\|\, \mathcal{U}) \,, \tag{39}$$

where $R(\mathbf{x})$ is the NLI entailment probability for the target topic and $\mathcal{U}$ the uniform prior. Since Adam is invariant to the overall loss scale, only the ratio $\lambda_{\text{KL}}/\beta$ matters.

Differentiable Bridge to the Reward Model. GIDD and the reward models use different tokenizers, so decoding to text and re-tokenizing would break gradient flow. We therefore precompute a fixed map from each GIDD token to a reward-model input embedding. Each GIDD token is first matched to a single reward-model token by string, accounting for different word-boundary markers (SentencePiece "_" vs. byte-level "Ġ"), stripped markers, lowercasing, and WordPiece continuations ("##"). If no match exists, the decoded token string is re-encoded with the reward tokenizer and the embeddings of the resulting tokens are averaged; remaining tokens map to the unknown token. The surrogate's per-position distribution p over GIDD tokens is then mapped to the expected embedding $\sum_v p(v)\,\bar{\mathbf{e}}(v)$,

yielding one reward-model embedding per generated position. These embeddings are combined with the hard embeddings of the task-specific context (here, the NLI hypothesis) and passed to the reward model through its input embeddings. The map covers the full GIDD vocabulary, so generation is not restricted. Since tokens that split into several reward-model tokens are collapsed into a single averaged embedding, the differentiable reward approximates, rather than exactly reproduces, the reward of the decoded and re-tokenized text. The held-out evaluator and GenPPL are computed on the decoded text.

G.7 Evaluation Protocol

Sample Generation. For each prompt, topic, and method, we produce 10 independent samples. For DDNO, each sample is an independent optimization run from freshly initialized noise logits; for Best-of-N and FK-Steering, each sample is the highest-reward output of an independent run.

Metrics. We report the following metrics, averaged over prompts and topics:

- **Reward**: NLI entailment probability under the optimization reward model (DeBERTa-v3-Large).
- **Evaluator**: entailment probability under the held-out evaluator (RoBERTa-Large-MNLI).
- **GenPPL**: GPT-2 XL perplexity of the generated text.
- **Entropy**: diversity entropy of the generated sequences.

Results are summarized in Figure 5.

H Open-Ended Response Quality Experiment Details

This section provides implementation details for the open-ended quality reward (QRM) experiments in Section 3.3. All experiments are conducted on a single H100 GPU.

H.1 Model and Architecture

We use the same model and sampling configuration as for topic steering (Section G): GIDD-Unif-3B (von Rütte et al., 2025), generating 64 tokens (excluding the prompt prefix) with 128 denoising steps, a cosine noise schedule, and adaptive sampling.

H.2 Task Configuration

We use 10 open-ended prompts that require multi-step reasoning, e.g., explaining the resource curse with three mechanisms, comparing climate–agriculture interactions in Egypt and Brazil, or explaining why Venus is hotter than Mercury despite being farther from the Sun. The full list is given in Table 14.

H.3 Reward and Evaluation Models

QRM (Optimization and Evaluation). We use QRM-Llama3.1-8B-v2 (Dorka, 2024), an 8B-parameter reward model based on Llama 3.1 that predicts the full reward distribution via quantile regression over 19 quantiles $(0.05, 0.10, \ldots, 0.95)$, with a learned gating network that weights attributes conditioned on the prompt. The scalar reward is the mean over the predicted quantiles. The generated text is formatted as a chat turn with the Llama 3.1 chat template before scoring: `[BOS] user: {prompt} assistant: {response} [EOS]`. Gradients reach the GIDD token distributions through the same differentiable bridge as in topic steering (Section G.6), with the chat-formatted prompt as hard context. Note that QRM is used both for optimization and for evaluation; GenPPL serves as the independent signal.

Perplexity Model. We measure fluency with GPT-2 XL (Radford et al., 2019) perplexity (GenPPL), computed on the decoded text.

Table 14: **Reasoning prompts** used in the QRM experiments.

#	Prompt
1	Why can a country be rich in natural resources and still remain poor? Explain with three mechanisms.
2	Why does increasing atmospheric CO_2 warm the Earth? Give a short explanation without equations.
3	Explain why Venus is hotter than Mercury, even though Mercury is closer to the Sun.
4	Why can antibiotics treat bacterial infections but not viral infections? Explain for a non-expert.
5	Why do some countries have many natural harbors while others have long coastlines with few major ports?
6	Compare why Singapore and Switzerland both became major economic hubs despite very different geography.
7	Compare how climate affects agriculture in Egypt and Brazil, and explain how that shapes their economies.
8	Why can access to the sea matter for economic development? Compare one landlocked country and one coastal country.
9	Explain why drought can increase both food prices and political instability.
10	A student asks why some diseases spread faster in cities than in rural areas. Explain using population density, mobility, and public health capacity.

H.4 Baseline: Best-of-N Sampling

For each prompt, we generate N independent samples from the GIDD base model with the same denoising configuration (adaptive sampling at temperature 0.85) and select the one with the highest QRM reward. Following the compute-accounting convention in Section E.3, we match T DDNO optimization steps with $N = 3T$ Best-of-N candidates. Both methods use 128 denoising steps per rollout. We repeat this for 5 independent groups per prompt, yielding 5 selected samples per prompt and budget.

H.5 DDNO Hyperparameters

Table 15: **DDNO hyperparameters for the QRM experiments.** Used for all 10 prompts.

Hyperparameter	Value
Optimization steps (T)	10 / 20 / 30 / 40 / 50
Best-of-N samples	N=$3T$
Learning rate	0.3
Guidance scale (β)	10 000
KL weight (λ_{KL})	0.1
Surrogate temperature (τ)	0.85
Initial logit scale (σ)	0.05
Gradient clip norm	0.0001
Sequence length (suffix)	64
Denoising steps	128
Samples per prompt	5

Optimization Details. We optimize the initial noise logits $\boldsymbol{\ell}_0 \in \mathbb{R}^{L\times V}$ (L=64, V the GIDD vocabulary size) with Adam (Kingma, 2014), initialized from $\mathcal{N}(0, 0.05^2 I)$ and clamped to $[-50, 50]$ for numerical stability. Each step backpropagates through the full 128-step denoising loop using GumbelSTE with fixed per-step Gumbel noise, with gradient checkpointing to reduce memory. We minimize

$$\mathcal{L} = -\beta \cdot R(\mathbf{x}) + \lambda_{\text{KL}} \cdot D_{\text{KL}}(\text{softmax}(\boldsymbol{\ell}_0) \,\|\, \mathcal{U}) \,, \tag{40}$$

where $R(\mathbf{x})$ is the QRM reward and $\mathcal{U}$ the uniform prior.

H.6 Evaluation Protocol

Sample Generation. For each prompt, method, and budget, we produce 5 samples. For DDNO, each sample is an independent optimization run from freshly initialized noise logits; for Best-of-N, each sample is the highest-reward output of an independent group of N base-model samples.

Metrics. We report the QRM reward of the final output and its GPT-2 XL perplexity, averaged over the 50 samples (10 prompts $\times$ 5). Results are summarized in Table 3 (three budgets, mean $\pm$ s.e.m.) and Table 16 (all budgets).

Table 16: **Full compute-matched QRM results** on the reasoning prompts (mean $\pm$ s.e.m. over 50 samples, i.e., 10 prompts $\times$ 5). T DDNO steps are matched with $N{=}3T$ Best-of-N samples.

		QRM ↑		GenPPL ↓	
T	N	Best-of-N	DDNO	Best-of-N	DDNO
Base		0.206 ± 0.021		40.22 ± 14.65	
10	30	0.458 ± 0.017	0.456 ± 0.018	14.44 ± 1.53	26.29 ± 10.34
20	60	0.500 ± 0.013	0.498 ± 0.016	15.72 ± 1.96	14.63 ± 3.03
30	90	0.518 ± 0.013	0.519 ± 0.015	13.86 ± 1.54	12.97 ± 2.42
40	120	0.528 ± 0.013	0.538 ± 0.015	13.54 ± 1.52	11.32 ± 0.56
50	150	0.538 ± 0.012	0.548 ± 0.015	14.42 ± 1.54	11.57 ± 0.60

I FURTHER RESULTS

I.1 QUALITATIVE IMAGE GENERATION

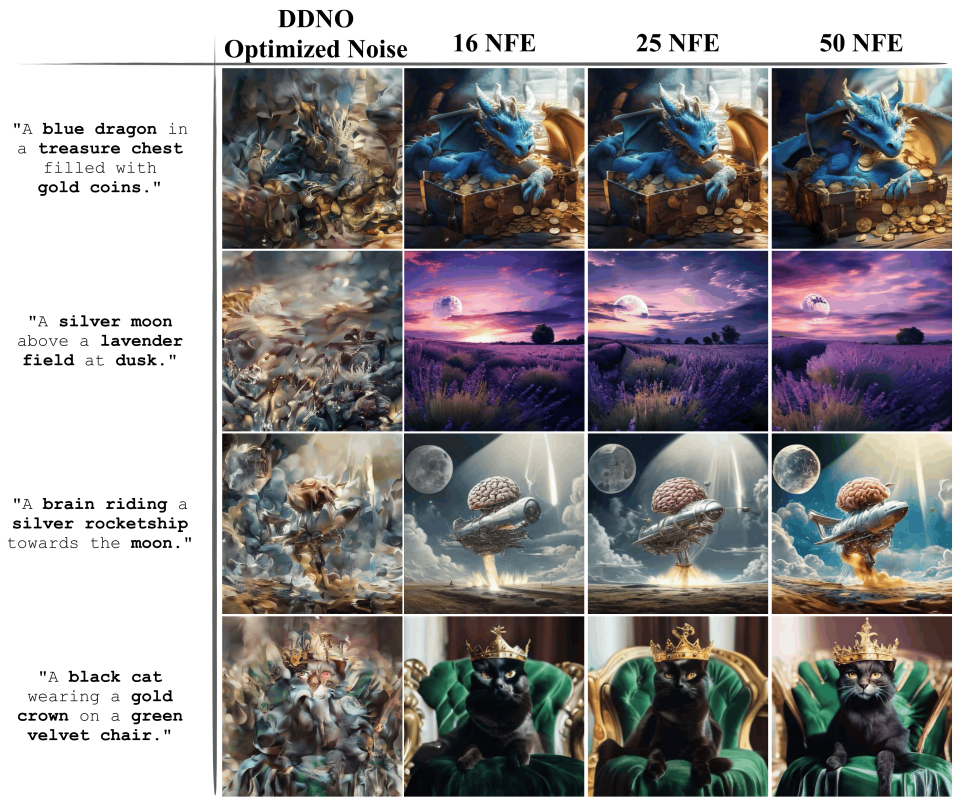

Figure 8: Results for DDNO optimized noise for different NFE. The optimization is done using 16 NFEs and we run generation from the same optimized noise with 16, 25, and 50 steps. The initial noise has a dominating effect on the generated image and generalizes to different NFEs during inference.

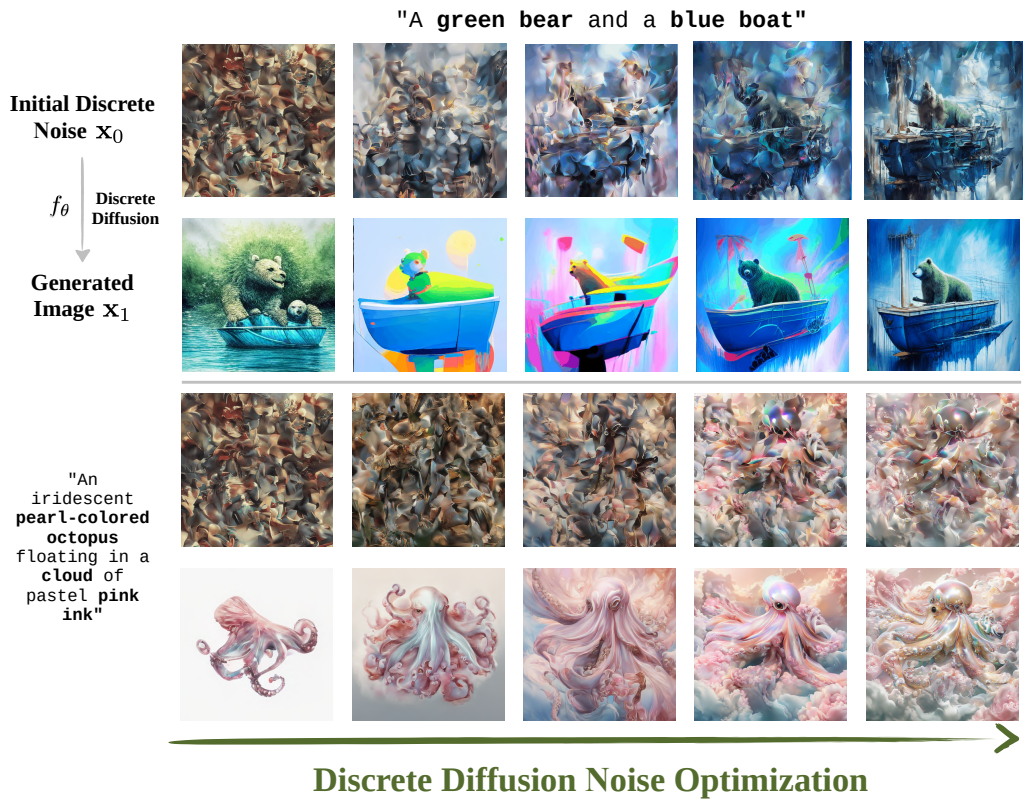

Figure 9: **Discrete Diffusion Noise Optimization (DDNO).** Our method enables high-fidelity, reward-guided generation by optimizing the initial source noise for a frozen discrete diffusion model f_θ. (Top) The optimization process begins with an initial noise sample $\mathbf{x}_0$. (Bottom) By propagating reward gradients through a differentiable unrolling of the diffusion process, **DDNO** refines the noise to generate samples $\mathbf{x}_1$ that precisely align with complex rewards, e.g. semantic constraints.

I.2 Qualitative Text Generation

Prompt: *"Someone asks you to explain how vaccines work to a skeptical family member. What would you say?"*		
Best-of-N	0.465	"You may need to explain that the vaccines are safe and effective, and that they are a form of prevention. You may also need to explain to your family that the vaccines are used to protect the population from certain diseases, and that they are not a cure for the —"
DDNO (Ours)	**0.627**	"It's simple. The vaccine is not a virus or a bacteria. It's a piece of protein. The vaccine tells the body to make the protein. And when it makes the protein, the immune system starts to attack the virus."

Figure 10: **Qualitative comparison with a text quality reward model.** DDNO produces a concrete, mechanism-level explanation; Best-of-N hedges and truncates.

Table 17: **Example generations for open-ended reasoning.** DDNO produces more complete and informative responses while achieving higher QRM reward.

Method	**Reward**	**Generated Text**
Prompt 1: "Why does increasing atmospheric CO2 warm the Earth? Give a short explanation without equations."		
Best-of-N	0.650	"It contributes to global warming by altering Earth's energy balance and climate. For instance, by absorbing more infrared radiation (IR), it increases Earth's surface temperature, leading to higher surface temperatures."
DDNO (Ours)	**0.834**	"Firstly, anthropogenic atmospheric CO2 is a greenhouse gas, meaning it traps heat in the atmosphere and warms the Earth's surface. Therefore, increasing atmospheric CO2 leads to global warming, as it increases the atmospheric CO2 concentration, leading to a rise in global temperature."
Prompt 2: "Someone asks you to explain how vaccines work to a skeptical family member. What would you say?"		
Best-of-N	0.465	"You may need to explain that the vaccines are safe and effective, and that they are a form of prevention. You may also need to explain to your family that the vaccines are used to protect the population from certain diseases, and that they are not a cure for the"
DDNO (Ours)	**0.627**	"It's simple. The vaccine is not a virus or a bacteria. It's a piece of protein. The vaccine tells the body to make the protein. And when it makes the protein, the immune system starts to attack the virus."

Table 18: **Example generations for sports topic steering.** Samples from different methods show varying degrees of topic adherence and fluency. Topic-relevant terms are underlined.

Method	**Generated Text**
Base Model	"On a gray Monday morning, Martin badges into the office a little early, coffee in hand, still buzzing from last night. He's aware he doesn't have the right job, but he can't stop thinking about it. He's met a lot of people in the past few weeks and he's spent a good amount of time looking..."
Best-of-50	"On a gray Monday morning, Martin badges into the office a little early, coffee in hand, still buzzing from last night. Today, he's sharing the desk with his friend mate, one of the top Djokovic soothed in the country, and his friend, phobes, who flew in from across the country..."
DDNO (Ours)	"On a gray Monday morning, Martin badges into the office a little early, coffee in hand, still buzzing from last night. He sits at his desk, fingers on the keyboard, a small smile on his face. He talks to a colleague, talking about the game, jazzing about inves,latures and terrain, quoting old tales. He watches the latest"

Table 19: **Example generations for science topic steering.** DDNO successfully steers generation toward scientific content, though at some cost to fluency.

Method	Generated Text
Base Model	"On a gray Monday morning, Martin badges into the office a little early, coffee in hand, still buzzing from last night. He settles into the chair, resting his head against the back of the chair. He glances down at the coffee table, noting that there's still a cup of coffee..."
Best-of-50	"On a gray Monday morning, Martin badges into the office a little early, coffee in hand, still buzzing from last night. He had hoped that he would be back on the team tomorrow, but he was, as usual, Viability and mological..."
DDNO (Ours)	"On a gray Monday morning, Martin badges into the office a little early, coffee in hand, still buzzing from last night. He has a dose to administer, a small dose of amigo prepared by Martin himself and administered by Mikes saliva. It's a small dose, Martin scans the length of the room, looking for the small white capsule. He finds the capsule, and"