DDNO: DISCRETE DIFFUSION NOISE OPTIMIZATION

Luca Eyring[1,*,†] **Vincent Pauline**[1,†] **Stefan Bauer**[1]
Alexey Dosovitskiy[2,‡] **Zeynep Akata**[1,‡]

[1]TU Munich, Helmholtz Munich, MCML [2]Inceptive
luca.eyring@tum.de vincent.paulinef@gmail.com

ABSTRACT

Aligning discrete diffusion models with downstream rewards remains challenging: step-wise guidance is myopic and degrades sample quality, while fine-tuning is expensive and task-specific. We introduce Discrete Diffusion Noise Optimization (DDNO), a training-free method that instead optimizes the initial discrete noise to maximize terminal rewards while keeping the generator frozen. DDNO parameterizes the noise distribution with continuous logits and propagates gradients through the reverse process via a straight-through surrogate combined with soft mixing, enabling stable optimization over long denoising trajectories. On compositional text-to-image synthesis and controllable text generation, DDNO consistently outperforms inference-time baselines like guidance and Best-of-N while exhibiting favorable scaling. This positions DDNO as a promising axis for test-time scaling in discrete generative models, complementing advances in continuous diffusion.

1 INTRODUCTION

Generative models are increasingly used not only to imitate data, but to solve *problems*: we want discrete sequences that are fluent, faithful, and useful according to a downstream objective. This is formalized as *reward-guided generation*, where a reward function $R(\mathbf{x})$ scores outputs for properties such as sentiment, constraint satisfaction, or human preference (Stiennon et al., 2020; Ziegler et al., 2019; Ouyang et al., 2022). The challenge lies in steering generation toward high-reward samples without sacrificing the coherence learned during training.

In continuous domains, the diffusion community has developed a rich toolbox for inference-time control: guidance mechanisms that steer the reverse process, as well as *noise optimization* procedures that directly optimize the initial noise to maximize terminal objectives (Ho & Salimans, 2022; Karunratanakul et al., 2024; Guo et al., 2024; Ben-Hamu et al., 2024; Eyring et al., 2024). These methods are appealing because they improve alignment without retraining the base model. Discrete diffusion and flow-matching models offer a promising foundation for extending these ideas to text, as they generate sequences non-autoregressively and can capture long-range dependencies essential for coherent language (Austin et al., 2021; Lou et al., 2023; Sahoo et al., 2024; Campbell et al., 2024).

Yet inference-time alignment for discrete diffusion remains challenging. Fine-tuning approaches update model parameters through reinforcement learning or preference optimization (Wang et al., 2024; Borso et al., 2025), but are computationally expensive and produce specialized checkpoints that sacrifice the versatility of general-purpose systems. Training-free guidance methods modify the sampling process while keeping weights frozen (Nisonoff et al., 2024; Schiff et al., 2025), but face a fundamental limitation: they operate through myopic, per-step corrections using local reward signals. This greedy approach creates two critical problems. First, repeated perturbations progressively push the sampling trajectory away from the learned data manifold, degrading sample quality. Second, step-wise optimization is inherently inefficient for maximizing terminal rewards, as it optimizes based on local improvements rather than the ultimate objective. The result is a difficult trade-off where pursuing higher rewards yields diminishing returns while substantially compromising sample fidelity.

[*]Work partially done during an internship at Inceptive
[†]Equal contribution
[‡]Equal advising

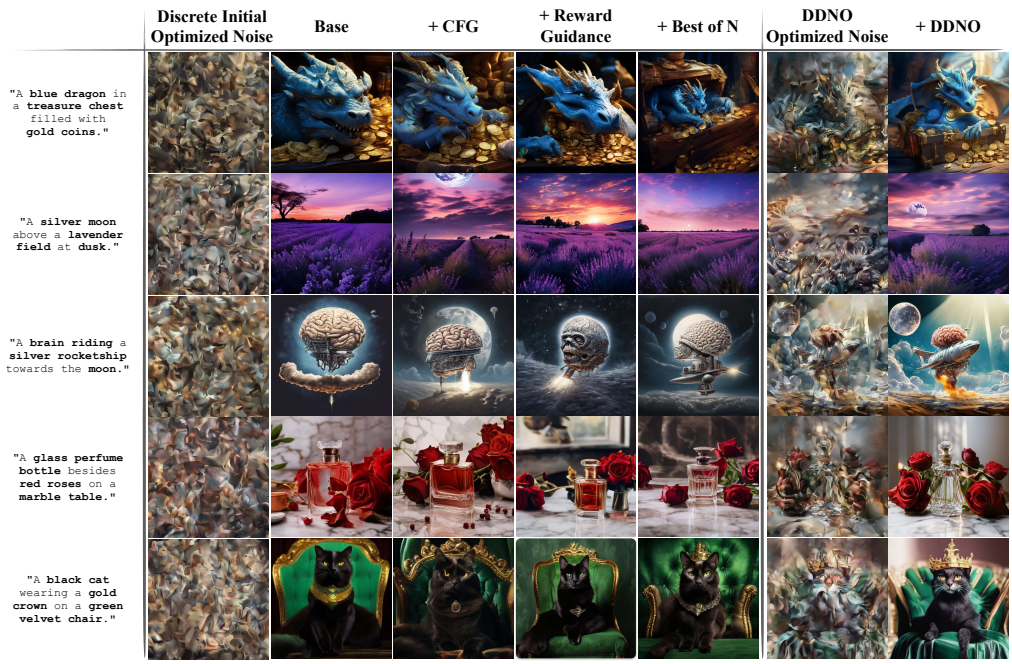

Figure 1: **Discrete Diffusion Noise Optimization (DDNO).** Qualitative comparison across five prompts for a frozen discrete diffusion model. **Left:** Existing inference-time methods (CFG, reward guidance, Best-of-N) applied to randomly sampled discrete initial noise. **Right: DDNO** optimizes the initial noise by propagating reward gradients through a differentiable unrolling of the reverse process. The optimized noise (rightmost columns) 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 itself, before any denoising is performed.

We take a different approach. Instead of modifying the sampling process, we optimize the starting point, i.e. searching for initial discrete noise sequence that naturally evolve into high-reward samples under the model's unchanged dynamics. Inspired by noise optimization successes in continuous diffusion (Wallace et al., 2023a; Ben-Hamu et al., 2024; Eyring et al., 2024), this reframes guidance as global optimization over initial conditions rather than a sequence of local corrections.

We introduce **Discrete Diffusion Noise Optimization (DDNO)**, a training-free inference-time method that parameterizes the initial noise distribution using continuous logits and optimizes these parameters to maximize terminal rewards. DDNO samples an initial discrete sequence from random logits ℓ_0, runs the frozen reverse process, and updates ℓ_0 via gradient descent. As the model dynamics are never altered, DDNO preserves the inductive biases learned during training while enabling trajectory-level optimization that eliminates the manifold deviation inherent to step-wise guidance.

The central technical obstacle is differentiability: discrete categorical sampling blocks gradients from terminal rewards back to the initial logits. We address this by developing a straight-through estimator (STE) surrogate that maintains discrete behavior in the forward pass, enabling gradient flow through a continuous relaxation in the backward pass (Jang et al., 2016; Maddison et al., 2016). This provides a stable, end-to-end learning signal without the approximation errors of alternative reparameterization techniques. Our main contributions can be summarized as

1. **Noise optimization for discrete diffusion:** We introduce DDNO, a training-free framework for test-time 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 preserves discrete dynamics while enabling end-to-end gradient flow.
3. **Empirical validation:** We demonstrate consistent improvements over guidance and Best-of-N baselines on compositional image generation and semantic text control tasks.

2 BACKGROUND

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.

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, hence $p_{\text{noise}}(\mathbf{x}) = K^{-D}$ for sequences. This choice is essential for DDNO. Under uniform noise, the initial sequence $\mathbf{x}_0$ has a profound effect on the final output: different initializations traverse different trajectories through the generative process, yielding different samples. DDNO exploits this structure 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 the initial *noise*. While masked diffusion has been the predominant paradigm, recent work demonstrates that uniform-noise models can match or exceed their performance (von Rütte et al., 2025; Schiff et al., 2025; Shaul et al., 2024; Sahoo et al., 2025), making this assumption increasingly practical.

Learning. The interpolation induces a continuous-time Markov chain (CTMC) governed by a rate matrix (Campbell et al., 2024; Lipman et al., 2024); standard derivations appear in Section C.2. The learnable component is a neural network f_θ that predicts the posterior over clean tokens given noisy states

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

the estimated probability that the clean token at position i is k.

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 $q_{s|t}^i(\cdot \mid \mathbf{x}_t)$. Two strategies instantiate this update. **Ancestral sampling** uses a *jump/stay* decomposition: for current token $c = \bar{x}_t^i$,

$$p_{\text{jump}}^i \coloneqq 1 - q_{s|t}^i(c \mid \mathbf{x}_t), \quad r_{s|t}^i(k) \coloneqq \frac{q_{s|t}^i(k \mid \mathbf{x}_t)\, \mathbf{1}\{k \neq c\}}{p_{\text{jump}}^i} \quad (\text{when } p_{\text{jump}}^i > 0). \tag{4}$$

The sampler stochastically draws a destination $y^i \sim \text{Cat}(r_{s|t}^i)$ and mask $m^i \sim \text{Bernoulli}(p_{\text{jump}}^i)$, 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), rather than stochastically updating all positions, deterministically selects the positions where the model is most confident a change is needed. Each position receives a score

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

where $\pi_\lambda(\bar{x}_t^i)$ is the prior probability of the current token under the noise distribution. The score σ^i is large when the model confidently prefers a different token; the sampler selects the top-k positions by score and overwrites them by sampling from $\hat{q}_{1|t}(\cdot \mid \mathbf{x}_t)$.

The Gradient Barrier. Across positions, these discrete update rules define a generative map $G_\theta : \mathcal{S} \to \mathcal{S}$. The discrete sampling operations block gradient propagation: given a reward $R : \mathcal{S} \to \mathbb{R}$, gradients cannot flow from $R(\mathbf{x}_1)$ back through G_θ, precluding direct optimization.

Inference-Time Alignment. Given a frozen generator G_θ and a reward function $R : \mathcal{S} \to \mathbb{R}$, inference-time alignment seeks individual samples $\mathbf{x}_1$ that maximize R without retraining the model. Existing guidance methods (Nisonoff et al., 2025; Schiff et al., 2025) modify the CTMC transition rates at every denoising step, but rely on plug-in approximations that degrade when posteriors are multimodal or rewards are non-smooth (Lee et al., 2025; Wan et al., 2025; Hasan et al., 2026); see Section C.4 for a formal treatment. These limitations motivate a different strategy: rather than approximating guided dynamics at each step, we optimize the *source* of the generative process to find an initial noise $\mathbf{x}_0$ from which G_θ naturally produces a high-reward output.

3 Discrete Diffusion Noise Optimization

We introduce **Discrete Diffusion Noise Optimization (DDNO)**, a training-free framework for inference-time reward alignment of discrete diffusion models. The central idea, inspired by noise optimization in continuous diffusion (Equation (37)), is to treat the initial noise sequence not as a fixed random draw but as a *control variable* optimized to steer generation toward high-reward outputs.

Let $G_\theta : \mathcal{S} \to \mathcal{S}$ denote the (stochastic) generative map that transforms initial noise $\mathbf{x}_0 \in \mathcal{S}$ into a final sample $\mathbf{x}_1$ via the learned reverse CTMC. 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. This *noise-as-control* perspective is the foundation of DDNO.

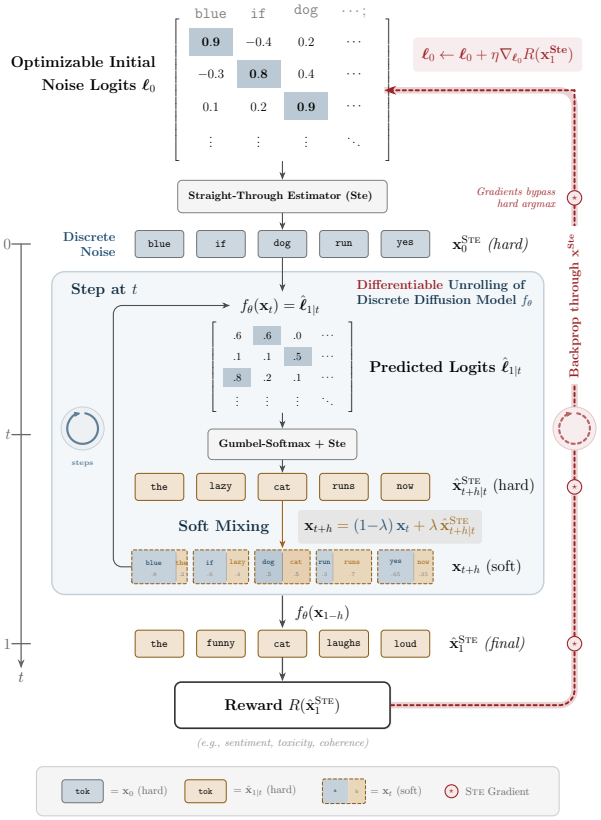

Figure 2: Sketch of **DDNO**, which optimizes the initial discrete noise sample $\mathbf{x}_0$ parametrized through logits ℓ_0 by propagating gradients through the differentiable surrogate unrolling of f_θ from a terminal reward $R(\mathbf{x}_1)$ back to ℓ_0.

3.1 Objective

Per-sample objective. Let $V(\mathbf{x}_0) := \mathbb{E}[R(\mathbf{x}_1) \mid \mathbf{x}_0]$ 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 C.4); the per-sample formulation equation 6 arises as its Dirac restriction under the uniform prior (Lemma C.4, Corollary C.5).

Optimization challenges. Equation 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 3.2), and a differentiable surrogate for the reverse dynamics (Section 3.3).

3.2 From Discrete Noise to Optimizable Logits

The initial state $\mathbf{x}_0$ and search space $\mathcal{S} = \{1, \ldots, K\}^D$ is discrete, precluding direct gradient-based optimization. We introduce a continuous parameterization that enables gradient flow while preserving compatibility with the pretrained model.

Noise logits. We parametrize the initial discrete state $\mathbf{x}_0$ through continuous logits $\ell_0 \in \mathbb{R}^{D \times K}$, which implicitly define a factorized categorical distribution

$$q_{0,\ell_0}(\mathbf{x}_0) = \mathrm{Cat}(\mathbf{x}_0 \mid \mathrm{softmax}(\ell_0)). \tag{7}$$

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

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 $t = 0, h, 2h, \ldots, 1-h$ (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: $\mathbf{x}_1$

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)

$$\boldsymbol{\epsilon} \sim \mathrm{Gumbel}(0,1)^K, \tag{8}$$

$$\mathbf{p}_{\boldsymbol{\epsilon}} = \mathrm{softmax}\big((\boldsymbol{\ell} + \boldsymbol{\epsilon})/\tau\big), \qquad \mathbf{h}_{\boldsymbol{\epsilon}} = \mathrm{onehot}\big(\arg\max_k (\ell_k + \epsilon_k)\big), \tag{9}$$

$$\textsc{GumbelSte}(\boldsymbol{\ell}, \tau) = \mathbf{p}_{\boldsymbol{\epsilon}} + \mathrm{sg}(\mathbf{h}_{\boldsymbol{\epsilon}} - \mathbf{p}_{\boldsymbol{\epsilon}}), \tag{10}$$

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 produce degenerate gradients that point equally in all directions. 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 $\mathbf{x}_0 \sim \mathrm{Uniform}([K])^{\otimes L}$, 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 3.3).

Entropy regularization. Although we decode via $\arg\max$ (a point estimate), the logits $\boldsymbol{\ell}_0$ implicitly define a factorized categorical $q_{0,\boldsymbol{\ell}_0}(\mathbf{x}_0) = \prod_i \mathrm{Cat}(x_0^i \mid \mathrm{softmax}(\boldsymbol{\ell}_0^i))$ through which gradients flow via Ste. Regularizing this implicit distribution toward the uniform prior gives

$$\mathrm{KL}(q_{0,\boldsymbol{\ell}_0} \| p_{\text{noise}}) = D \log K - \sum_{i=1}^{D} H\big(\mathrm{softmax}(\boldsymbol{\ell}_0^i)\big), \tag{11}$$

where $H(\cdot)$ denotes entropy. Minimizing this KL—equivalently, maximizing entropy—serves two purposes: (i) it instantiates the variational regularizer from Equation (30) for our relaxed parameterization, and (ii) it prevents logit saturation, ensuring Ste gradients remain well-conditioned.

Objective. Combining these components, the DDNO test-time objective reads

$$\boldsymbol{\ell}_0^\star = \arg\max_{\boldsymbol{\ell}_0} \; R\big(G_\theta(\textsc{Ste}(\boldsymbol{\ell}_0, \tau))\big) - \frac{1}{\beta} \sum_{i=1}^{D} H\big(\mathrm{softmax}(\boldsymbol{\ell}_0^i)\big). \tag{12}$$

We aim to optimize this via gradient ascent on $\boldsymbol{\ell}_0$. At this stage, we have a continuous parameterization, a differentiable sampling mechanism, and a regularized objective. However, the generative process G_θ itself remains non-differentiable due to categorical sampling at each step. Importantly, the machinery developed above is *sampler-agnostic*: it parameterizes and regularizes the initial noise independently of how the reverse process advances.

3.3 DIFFERENTIABLE SURROGATE PROCESS

The reverse process G_θ 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 that 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 $\mathbf{x}_{s|t} \leftarrow \textsc{GumbelSTE}(\log \mathbf{r}_{s|t}, \tau)$. 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

$$\mathbf{x}_s = (\mathbf{1} - \mathbf{m}) \odot \mathbf{x}_t + \mathbf{m} \odot \mathbf{x}_{s|t}. \tag{13}$$

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

$$\mathbf{x}_s = (\mathbf{1} - \mathbf{p}_{\mathrm{jump}}) \odot \mathbf{x}_t + \mathbf{p}_{\mathrm{jump}} \odot \mathbf{x}_{s|t}. \tag{14}$$

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.

Variance Reduction via Rao-Blackwellization. The substitution $\mathbf{m} \to \mathbf{p}_{\mathrm{jump}}$ 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. Concretely

$$\mathbb{E}[\mathbf{x}_s \mid \mathbf{x}_{s|t}] = \mathbb{E}[\mathbf{1} - \mathbf{m} \mid \mathbf{x}_{s|t}] \odot \mathbf{x}_t + \mathbb{E}[\mathbf{m} \mid \mathbf{x}_{s|t}] \odot \mathbf{x}_{s|t} = (\mathbf{1} - \mathbf{p}_{\mathrm{jump}}) \odot \mathbf{x}_t + \mathbf{p}_{\mathrm{jump}} \odot \mathbf{x}_{s|t}, \tag{15}$$

since $\mathbf{m} \perp \mathbf{x}_{s|t}$ and $\mathbb{E}[\mathbf{m}] = \mathbf{p}_{\mathrm{jump}}$. By the law of total variance, this conditioning preserves the mean while strictly reducing variance at the state-estimator level (Appendix D.2).

The combined effect is a surrogate that maintains discrete tokens in the forward pass while enabling smooth gradient flow: GumbelSTE provides gradients through destination selection, and soft mixing provides gradients through jump probabilities. Algorithm 1 summarizes the complete procedure.

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

$$\mathbf{x}_s = \mathbf{x}_t + \mathrm{sg}(\mathbf{U}_t) \odot (\mathbf{x}_{\mathrm{prop}} - \mathbf{x}_t), \qquad U_t^i = \mathbf{1}[i \in \mathcal{I}_t], \tag{16}$$

where $\mathcal{I}_t = \mathrm{TopK}(\mathrm{sg}(\boldsymbol{\sigma}_t), m)$ 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 setting.

3.4 DETERMINISTIC SURROGATE PROCESS

The surrogate of Section 3.3 introduces Gumbel noise $\boldsymbol{\epsilon}_t^i$ at every denoising step to implement GUMBELSTE. If these samples are redrawn at each gradient step, identical logits $\boldsymbol{\ell}_0$ can produce wastly 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 thus preserves the exploration provided by categorical sampling while providing a smooth, reproducible loss landscape for gradient ascent.

4 Related Work

Reward Fine-tuning. Reward-alignment for Discrete Diffusion Models has been mostly tackled at training-time by fine-tuning the model. Approaches include preference optimization (Borso et al., 2025), RL-based approaches (Zhao et al., 2025; Marion et al., 2025; Rector-Brooks et al., 2025; Wang et al., 2025a), or directly using gradients of rewards to finetune the model. Specifically, DRAKES (Wang et al., 2024) also employs Gumbel-Softmax to obtain a differentiable generation, and directly updates model parameters with the obtained gradients. Continuous diffusion models have also been successfully aligned with rewards using reinforcement learning (Black et al., 2024; Fan et al., 2023; Deng et al., 2024; Zhang et al., 2024; Chen et al., 2024; Venkatraman et al., 2025), or direct reward fine-tuning (Lee et al., 2023; Li et al., 2024; Prabhudesai et al., 2023; Xu et al., 2023; Clark et al., 2023; Domingo-Enrich et al., 2024; Jena et al., 2024). Another line of work (Eyring et al., 2025; Wagenmaker et al., 2025; Venkatraman et al., 2025; Carter et al., 2026) closely related to DDNO trains separate noise hypernetworks to output reward aligned initial noise samples.

Test-time Reward Alignment. Test-time methods for continuous diffusion improve generation by finding better initial noise or refining intermediate states under reward guidance. These divide into search-based approaches (Ma et al., 2025; Uehara et al., 2025a;b; Karthik et al., 2023) that evaluate multiple candidates, and optimization-based approaches (Wallace et al., 2023b; Ben-Hamu et al., 2024; Novack et al., 2024; Karunratanakul et al., 2024; Guo et al., 2024; Tang et al., 2024; Eyring et al., 2024) that refine noise via gradient descent. Existing test-time methods for discrete diffusion focus on guidance (Nisonoff et al., 2025; Schiff et al., 2025), which requires approximations that degrade under multimodal posteriors (Section 2). Building up on this, DDNO is, to our knowledge, the first noise-optimization framework for discrete diffusion models.

5 Experiments

Our experiments address three 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 text and image generation. For images, we benchmark general preference alignment in compositional text-to-image generation while for text we focus on semantic control tasks. Throughout, all model parameters remain frozen; only the initial noise logits are optimized. Section 5.1 validates our surrogate design, Section 5.2 presents image generation results, and Section 5.3 presents text generation results.

5.1 Surrogate Design Ablation

Before evaluating DDNO end-to-end, we ablate the surrogate design along two axes: (i) *mixing*—whether the jump decision uses soft convex combinations (Equation (14)) or hard Bernoulli masks (Equation (13)), and (ii) *sampling*—whether the forward pass propagates soft probabilities or hard tokens via straight-through estimation. The key trade-off is fidelity to the original sampler versus differentiability for gradient-based optimization.

Table 1: Ablation of surrogate design choices. While hard mixing yields higher raw fidelity, it introduces significant gradient instability due to discrete mixing destroying the gradient flow. Results are obtained with 50 NFE.

Configuration	GenEval Mean	Opt.
Soft mixing + Soft sampling	0.60	✓
Soft mixing + Hard sampling (Ours)	**0.67**	✓
Hard mixing + Hard sampling (FUDOKI Base)	0.72	✗

Table 1 reports GenEval scores on FUDOKI. Soft mixing with soft sampling incurs the largest fidelity drop (0.60 vs. 0.72 base) due to distribution shift: the model was trained on discrete tokens, not continuous mixtures. Hard mixing with hard sampling better preserves fidelity (0.72) but blocks gradient flow entirely. Our design—soft mixing with hard sampling—achieves competitive fidelity (0.67) while remaining fully differentiable. We find that even a few consecutive steps of hard mixing cause gradient signal to vanish rapidly, motivating our soft mixing approach.

Table 2: **GenEval results for compositional text-to-image synthesis.** DDNO substantially outperforms all baselines, including Best-of-N, demonstrating effective gradient-based steering of discrete diffusion models toward compositional objectives.

Model	Mean ↑	Single↑	Two↑	Counting↑	Colors↑	Position↑	Attribution↑
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
FUDOKI Base (Wang et al., 2025b)	0.76	0.98	0.83	0.56	0.89	0.68	0.64
+ CFG (Ho & Salimans, 2022)	0.77	0.96	0.87	0.57	0.90	0.67	0.66
+ Reward Guidance (Nisonoff et al., 2025)	0.78	0.98	0.75	0.64	0.97	0.66	0.67
+ Best-of-N (Karthik et al., 2023)	0.88	0.98	0.95	0.73	0.94	0.88	0.78
FUDOKI Differentiable Surrogate	0.67	0.98	0.73	0.45	0.84	0.56	0.43
+ DDNO (Ours)	0.93	1.00	0.96	0.88	0.96	0.84	0.91

5.2 Image Generation: General Preference Alignment

Setup. We use FUDOKI (Wang et al., 2025b), a discrete flow matching model with kinetic-optimal paths (Shaul et al., 2024). We evaluate on GenEval (Ghosh et al., 2023), which measures compositional understanding across six categories: single/two object presence, counting, colors, position, and attribute binding. The reward combines semantic fidelity (NVILA-Lite-2B-Verifier logits) with aesthetic quality (HPSv2.1): $R = R_{\text{fidelity}} + 0.5 \cdot R_{\text{aesthetic}}$. We optimize noise logits for 50 steps using Adam with learning rate 0.5. We compare against CFG (Ho & Salimans, 2022), reward guidance (Nisonoff et al., 2025), and best-of-N sampling (Karthik et al., 2023). For the full implementation details, we refer to Appendix E.

Results. Table 2 presents the main results. DDNO achieves a mean GenEval score of 0.93, improving over the surrogate baseline (0.67) by 39% relative and over the original FUDOKI sampler (0.76) by 22%. This demonstrates that optimizing the initial noise distribution unlocks latent compositional capabilities within the frozen model. DDNO substantially outperforms all inference-time baselines: CFG (0.77), reward guidance (0.78), and best-of-N (0.88), despite the latter generating the same number of images ($N = 50$). As one step of DDNO is ≈3x more expensive, we compute-match Best-of-N and DDNO by performing the optimization with only 16 NFE instead of the 50 used for Best-of-N. We plot the compute-matched performance compared to Best-of-N in Figure 3. Figure 6 corroborates these results with qualitative examples of DDNO and competing methods.

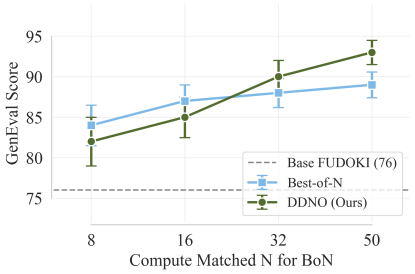

Figure 3: *Compute-matched* GenEval scores comparing Best-of-N to DDNO. Best-of-N is preferable at low compute budgets, but DDNO scales more efficiently and surpasses it at higher N.

5.3 Text Generation

Setup. We evaluate DDNO on controllable text generation, steering a frozen discrete diffusion language model toward target semantic attributes while maintaining fluency. We use GIDD-Unif-3B (von Rütte et al., 2025), a 3B parameter discrete diffusion model with uniform noise initialization and adaptive sampling throughout. We consider two different *topic steering* control tasks, where we optimize toward specific topics. To this end, we leverage a DeBERTa-v3-Large fine-tuned on multiple NLI datasets (Lewis et al., 2020; Laurer et al., 2024; Williams et al., 2018; He et al., 2020), computing reward as the entailment probability for the hypothesis "This text is about {topic}." To ensure disentangled evaluation, we measure evaluation scores with a held-out classifier (RoBERTa-Large-MNLI) not used during optimization (Liu et al., 2019). We evaluate on 10 diverse prompts with 10 samples per prompt and report generation perplexity (GenPPL, computed with GPT2-XL), reward scores (R) from the optimization objective, and evaluator scores (E) from the held-out

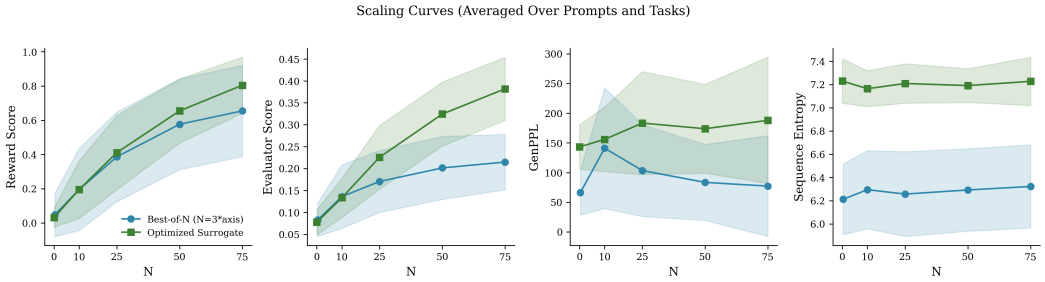

Figure 4: **Scaling comparison on topic steering.** DDNO vs Best-of-N *compute-matched* comparison of (i) optimization reward, (ii) held-out evaluator scores, (iii) GenPPL, and (iv) entropy.

Table 3: Reward optimization with a text quality reward comparing DDNO and Best-of-N. *QRM Reward* reports the best-along-trajectory reward (mean ± std across prompts). *PPL Init* is the perplexity of the initial generation before optimization or selection. Bold indicates the highest reward per category. DDNO outperforms Best-of-N w.r.t. both reward and GenPPL.

		QRM Reward	QRM Init	GenPPL (↓)	GenPPL Init	Entropy
Standard	BoN (n=45)	0.506 ± 0.071	-	14.7 ± 4.6	238.5 ± 120.1	6.11 ± 0.19
	Opt (T=15)	0.506 ± 0.055	0.339 ± 0.034	14.7 ± 6.5	166.8 ± 125.1	6.17 ± 0.20
	Opt (T=25)	**0.539 ± 0.050**	0.338 ± 0.035	**12.7 ± 3.1**	191.2 ± 137.6	6.09 ± 0.20
Difficult	BoN (n=45)	0.483 ± 0.093	-	17.3 ± 6.3	178.1 ± 68.5	5.93 ± 0.21
	Opt (T=15)	0.510 ± 0.066	0.355 ± 0.054	15.3 ± 3.5	298.3 ± 286.1	5.94 ± 0.21
	Opt (T=25)	**0.544 ± 0.086**	0.351 ± 0.051	**14.2 ± 3.8**	278.7 ± 287.9	5.86 ± 0.25
All	BoN (n=45)	0.494 ± 0.084	-	16.0 ± 5.6	208.3 ± 102.3	6.02 ± 0.22
	Opt (T=15)	0.508 ± 0.061	0.347 ± 0.046	15.0 ± 5.2	232.5 ± 230.4	6.06 ± 0.24
	Opt (T=25)	**0.542 ± 0.071**	0.345 ± 0.044	**13.5 ± 3.5**	235.0 ± 229.8	5.98 ± 0.25

classifier. We compare against the base model, Best-of-N sampling (Karthik et al., 2023), Reward-Guidance (Nisonoff et al., 2024; Schiff et al., 2025), and an unoptimized surrogate baseline.

Scaling Behavior. Figure 4 illustrates the scaling characteristics of DDNO versus Best-of-N sampling on both topic steering tasks. The two approaches exhibit fundamentally different scaling regimes. Best-of-N sampling shows sublinear scaling with diminishing returns: the probability of sampling high-quality outputs through random search diminishes rapidly for challenging semantic targets, and increasing N yields marginal improvements once the easy samples are exhausted. DDNO, in contrast, demonstrates consistent improvement with additional optimization iterations, as each step contributes meaningful gradient signal toward the reward objective and thus, better scaling.

Reasoning Task. We further evaluate on open-ended reasoning, replacing the NLI classifier with a general-purpose quality reward model (QRM) evaluted over 20 prompts of varying difficulty with 10 evaluation samples per prompt. As shown in Table 3, DDNO with T=15 and T=25 outperforms Best-of-45 in QRM reward (0.542 vs. 0.494) and perplexity (13.5 vs. 16.0), with gains most pronounced on difficult prompts (+12.6%). Example qualitative texts are provided in Table 4.

6 Discussion

We introduced DDNO, a test-time optimization framework for reward alignment of discrete diffusion models. By parameterizing initial noise through continuous logits and constructing a differentiable surrogate via GumbelSTE and soft mixing, DDNO enables gradient-based optimization over the combinatorially large space of initial sequences. Experiments on compositional image generation and semantic text control demonstrate consistent improvements over guidance and Best-of-N.

7 ACKNLOWEDGEMENTS

We thank Henning Meyer, Tibor Rothschild, Alex Hawkins-Hooker and Rico Jonschkowski for insightful discussions and feedback. Zeynep Akata acknowledges funding by the ERC (853489 - DEXIM) and the Alfried Krupp von Bohlen und Halbach Foundation. Luca Eyring would like to thank the European Laboratory for Learning and Intelligent Systems (ELLIS) PhD program for support. Luca Eyring is supported by a Google PhD Fellowship in Machine Learning. This work was partially supported by the Munich Center for Machine Learning (MCML).

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.

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.

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.

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, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

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

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.

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.

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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 7871–7880, 2020.

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*, 2024.

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.

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.

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.

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.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

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, 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, 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. *arXiv preprint arXiv:2311.12908*, 2023a.

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

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.

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.

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 ALGORITHMS

A.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:

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

Adaptive position selection (shared):

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

$\mathcal{I}_t \leftarrow \mathrm{TopK}(\Delta_t, m)$ *// 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)

A.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*
  $J(\boldsymbol{\ell}_0) \leftarrow R(\mathbf{x}_1^{\text{ste}}) + \mathrm{KL}(p_{\boldsymbol{\ell}_0} \| p_{noise})$
  $\boldsymbol{\ell}_0 \leftarrow \boldsymbol{\ell}_0 + \eta \nabla_{\boldsymbol{\ell}_0} J(\boldsymbol{\ell}_0)$
  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: $\mathbf{x}_1^{\text{best}}$

B FURTHER RESULTS

B.1 QUALITATIVE IMAGE GENERATION

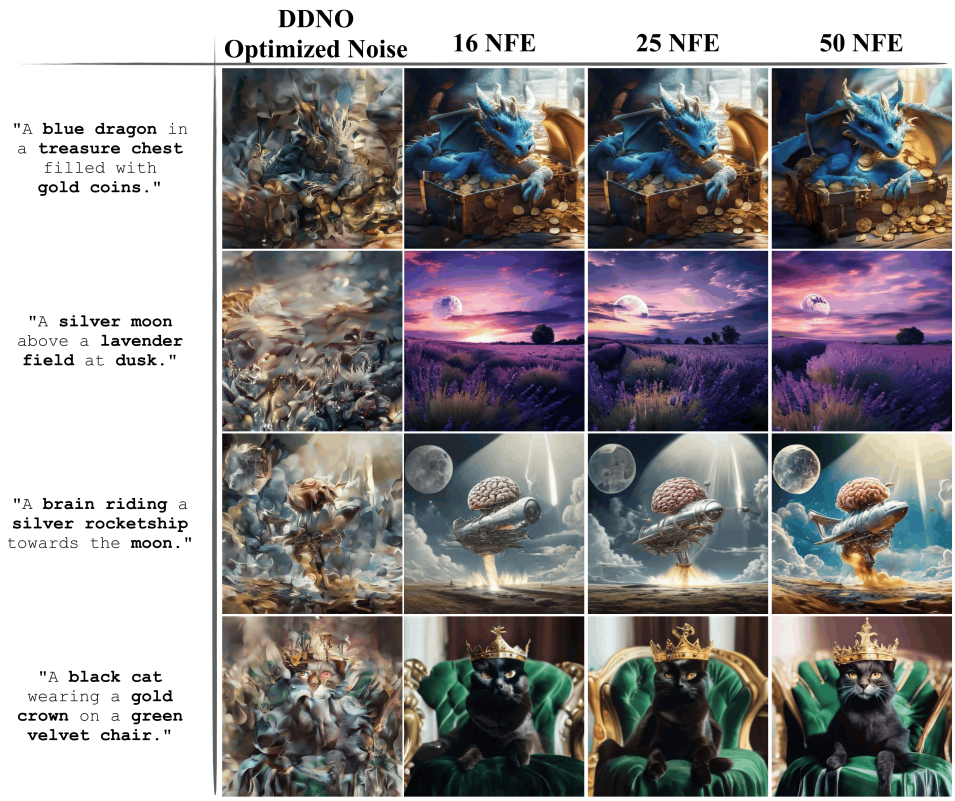

Figure 5: Results for DDNO optimized noise for different NFE. The optimization is done using 16 NFEs and we run generation from the same optimized noise with 16, 25, and 50 steps. The initial noise has a dominating effect on the generated image and generalizes to different NFEs during inference.

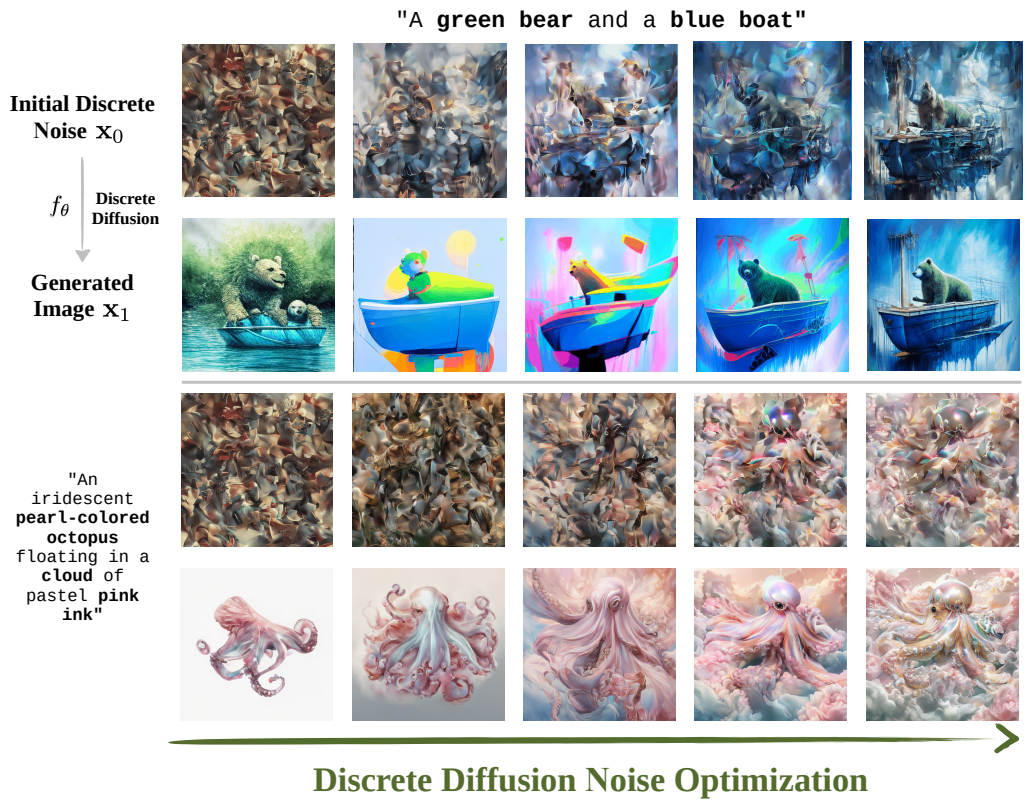

Figure 6: **Discrete Diffusion Noise Optimization (DDNO).** Our method enables high-fidelity, reward-guided generation by optimizing the initial source noise for a frozen discrete diffusion model f_θ. (Top) The optimization process begins with an initial noise sample $\mathbf{x}_0$. (Bottom) By propagating 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.

B.2 Qualitative Text Generation

Table 4: **Example generations for open-ended reasoning.** DDNO produces more complete and informative responses while achieving higher QRM reward. Prompts are drawn from the 10-prompt evaluation set.

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 5: **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 6: **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"

C Theoretical Foundations

C.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 $\frac{d\mathcal{P}}{d\mathcal{Q}}$.

C.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:

$$q_0(\mathbf{x}) = p_{\text{noise}}(\mathbf{x}), \qquad q_1(\mathbf{x}) = p_{\text{data}}(\mathbf{x}). \tag{17}$$

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:

$$q_t(\mathbf{x}|\mathbf{x}_1) = \prod_{i=1}^{D} q_t(x^i|x_1^i). \tag{18}$$

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

$$q_t(x^i|x_1^i) = (1 - \alpha_t)p_{\text{noise}}(x^i) + \alpha_t \delta_{x_1^i}(x^i), \quad \alpha_0 = 0, \alpha_1 = 1, \tag{19}$$

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):

$$q_{t+h|t}(\mathbf{x}|\mathbf{z}) = \delta_{\mathbf{z}}(\mathbf{x}) + h\, u_t(\mathbf{x}, \mathbf{z}) + o(h), \tag{20}$$

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),

$$u_t(\mathbf{x}, \mathbf{y}) = \frac{\partial}{\partial s}\, q_{s|t}(\mathbf{y} \mid \mathbf{x})\Big|_{s=t^+}, \qquad \mathbf{x} \neq \mathbf{y}. \tag{21}$$

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

$$u_t(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{D} u_t^i(x^i, \mathbf{z}) \prod_{j \neq i} \delta_{z^j}(x^j), \tag{22}$$

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

$$u_t^i(x^i, \mathbf{z}) = \sum_{x_1^i \in \mathcal{T}} u_t^i(x^i, z^i | x_1^i)\, q_{1|t}^i(x_1^i|\mathbf{z}), \tag{23}$$

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

$$\hat{q}_{1|t}(x_1^i = k \mid \mathbf{x}_t, t) = \mathrm{softmax}(\hat{\boldsymbol{\ell}}_{1|t}^i)_k, \tag{24}$$

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 (20), (22) and (23). A convenient equivalent implementation samples via a *jump/stay* decomposition: let $c = \bar{x}_t^i$ be the current token,

$$p_{\text{jump}}^i \;\coloneqq\; 1 - q_{s|t}^i(c \mid \mathbf{x}_t), \tag{25}$$

$$r_{s|t}^i(k) \;\coloneqq\; \frac{q_{s|t}^i(k \mid \mathbf{x}_t)\, \mathbf{1}\{k \neq c\}}{p_{\text{jump}}^i} \quad (\text{if } p_{\text{jump}}^i > 0), \tag{26}$$

then sample a destination $y^i \sim \mathrm{Cat}(r_{s|t}^i)$ and a mask $m^i \sim \mathrm{Bernoulli}(p_{\text{jump}}^i)$, 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 C.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.

C.3 Sampling via jump/stay decomposition

Lemma C.1 (Jump decomposition equals the categorical transition)**.** *Let $q \in \Delta^{K-1}$ and $c \in \{1, \ldots, K\}$. Define $p_{\text{jump}} = 1 - q(c)$ and $r(k) = \frac{q(k)\mathbf{1}\{k \neq c\}}{p_{\text{jump}}}$ for $p_{\text{jump}} > 0$. Let $m \sim \mathrm{Bernoulli}(p_{\text{jump}})$ 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. For $k = c$, $\mathbb{P}(x = c) = \mathbb{P}(m = 0) = 1 - p_{\text{jump}} = q(c)$. For $k \neq c$, $\mathbb{P}(x = k) = \mathbb{P}(m = 1)\mathbb{P}(y = k) = p_{\text{jump}} r(k) = q(k)$. □

C.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

$$q_\beta(\mathbf{x}) \;=\; \frac{q(\mathbf{x})\, e^{\beta R(\mathbf{x})}}{Z_\beta}, \qquad Z_\beta \;=\; \mathbb{E}_{\mathbf{x}\sim q}\big[e^{\beta R(\mathbf{x})}\big], \tag{27}$$

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)

$$\mathcal{P}^\star = \arg\min_{\mathcal{P}} \Big\{ D_{\mathrm{KL}}(\mathcal{P}\|\mathcal{Q}) - \beta\, \mathbb{E}_{\mathcal{P}_1}[R(\mathbf{x}_1)] \Big\}, \qquad q_\beta = \mathcal{P}_1^\star. \tag{28}$$

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

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):

$$u_{t,\beta}(\mathbf{x}, \mathbf{z}) = u_t(\mathbf{x}, \mathbf{z})\, \frac{\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x})}[e^{\beta R(\mathbf{x}_1)}]}{\mathbb{E}_{q(\mathbf{x}_1|\mathbf{z})}[e^{\beta R(\mathbf{x}_1)}]}, \qquad \mathbf{x} \neq \mathbf{z}, \tag{29}$$

where $u_t(\mathbf{x}, \mathbf{z})$ is the CTMC transition rate from state $\mathbf{x}$ to $\mathbf{z}$ (Section C.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. DDNO avoids modifying the transition dynamics altogether. Optimizing only the initial distribution q_0 while keeping pretrained transitions unchanged reduces the path-space KL to a divergence on initial conditions alone:

$$q_0^\star = \arg\max_{q_0} \; \Big\{ \mathbb{E}_{\mathbf{x}_0\sim q_0}\big[V(\mathbf{x}_0)\big] - \tfrac{1}{\beta}\, \mathrm{KL}(q_0\|p_{\text{noise}}) \Big\}, \tag{30}$$

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 C.4, Corollary C.5).

C.5 Tilted Distribution and Variational Characterization

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

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

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

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

$$D_{KL}(\mathcal{P}\|\mathcal{P}_\beta) = \mathbb{E}_{\mathcal{P}}\left[\log \frac{d\mathcal{P}}{d\mathcal{P}_\beta}\right] = \mathbb{E}_{\mathcal{P}}\left[\log \frac{d\mathcal{P}}{d\mathcal{Q}} \cdot \frac{d\mathcal{Q}}{d\mathcal{P}_\beta}\right] \tag{32}$$

$$= \mathbb{E}_{\mathcal{P}}\left[\log \frac{d\mathcal{P}}{d\mathcal{Q}}\right] + \mathbb{E}_{\mathcal{P}}\left[\log \frac{Z_\beta}{e^{\beta R(\mathbf{x}_1)}}\right] \tag{33}$$

$$= D_{KL}(\mathcal{P}\|\mathcal{Q}) - \beta\, \mathbb{E}_{\mathcal{P}}[R(\mathbf{x}_1)] + \log Z_\beta. \tag{34}$$

□

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

$$\mathcal{P}_\beta = \arg\min_{\mathcal{P}} \left\{ D_{KL}(\mathcal{P}\|\mathcal{Q}) - \beta\, \mathbb{E}_{\mathcal{P}}[R(\mathbf{x}_1)] \right\}. \tag{35}$$

C.6 Stochastic Optimal Control Perspective

The path-space variational problem equation 30 is an instance of *KL-regularized stochastic optimal control* (Levine, 2018; Bhole et al., 2025; Berner et al., 2022). In this framework, an agent seeks to maximize expected reward while paying a cost for deviating from reference dynamics:

$$\max_{\mathcal{P}} \left\{ \mathbb{E}_{\mathcal{P}}[R(\mathbf{x}_1)] - \frac{1}{\beta} \mathrm{KL}(\mathcal{P} \| \mathcal{Q}) \right\}. \tag{36}$$

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)

C.7 Continuous Test-Time Noise Optimization.

Test-time noise optimization Wallace et al. (2023b); Ben-Hamu et al. (2024); Novack et al. (2024); Karunratanakul et al. (2024); Guo et al. (2024); Tang et al. (2024); Eyring et al. (2024) 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

$$\mathbf{x}_0^\star = \arg\max_{\mathbf{x}_0} (r(g_\theta(\mathbf{x}_0)) - \mathrm{Reg}(\mathbf{x}_0)), \tag{37}$$

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.

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

Lemma C.4 (Dirac restriction yields MAP objective)**.** *Restricting Equation* (30) *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 C.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 C.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 30, and prevents logit saturation that would otherwise degrade gradient quality.

D Proofs

D.1 Minimum MSE estimator

We note X_t a random variable and $\mathcal{F}_t$ it's filtration. The MSE estimator of X_t is $\mathbb{E}[X_t \mid \mathcal{F}_t]$.

Proof. We consider the space of random variables admitting a finite second moment, $L^2(\mathcal{T}, \mathcal{F}_t, \mathbb{P}_t)$. On this space, the conditional expectation w.r.t. the filtration $\mathcal{F}_t$ corresponds to the orthogonal projection over $\mathcal{F}_t$-measurable variables. By applying Pythagora theorem, for all H $\mathcal{F}_t$-measurable, we have:

$$X_t - H = X_t - \mathbb{E}[X_t \mid \mathcal{F}_t] + \mathbb{E}[X_t \mid \mathcal{F}_t] - H \tag{38}$$

$\mathbb{E}[X_t \mid \mathcal{F}_t] - H$ is $\mathcal{F}_t$-measurable so we can apply Pythagora theorem in L^2.

$$\mathbb{E}[\|X_t - H\|^2] = \mathbb{E}[\|X_t - \mathbb{E}[X_t \mid \mathcal{F}_t]\|^2] + \mathbb{E}[\|\mathbb{E}[X_t \mid \mathcal{F}_t] - H\|^2] \tag{39}$$

Thus:

$$\text{MSE}_{X_t}(H) \geq \text{MSE}_{X_t}(\mathbb{E}[X_t \mid \mathcal{F}_t]) \tag{40}$$

□

D.2 Rao-Blackwellization of the mask gated update

Proposition D.1. *By the law of total variance and the tower property (since $\mathcal{G}_0 \subseteq \mathcal{G}_1$), our Rao-Blackwellized objective lowers the conditional sampling variance of X_s w.r.t. $\mathcal{G}_0$ compared to the Bernoulli gated base update*

$$\text{Var}\big(\mathbb{E}[X_s^i \mid \mathcal{G}_1] \,\big|\, \mathcal{G}_0\big) \;\leq\; \text{Var}\big(X_s^i \,\big|\, \mathcal{G}_0\big), \qquad \mathbb{E}[\mathbb{E}[X_s^i \mid \mathcal{G}_1]^i \mid \mathcal{G}_0] = \mathbb{E}[X_s^i \mid \mathcal{G}_0], \tag{41}$$

with strict inequality unless degeneracy case.

Proof. We fix an index i and write X_s^i the one-step update at that position. We assume $X_s \in L^2$, and $\mathcal{G}_0 \subseteq \mathcal{G}_1$.

Mean preservation (tower property). Since $\mathcal{G}_0 \subseteq \mathcal{G}_1$,

$$\mathbb{E}[\mathbb{E}[X \mid \mathcal{G}_1] \mid \mathcal{G}_0] = \mathbb{E}[X \mid \mathcal{G}_0]. \tag{42}$$

Define the $\mathcal{G}_0$-conditional mean $\mathbb{E}[X \mid \mathcal{G}_0]$ and the centered variable $W \coloneqq X - \mathbb{E}[X \mid \mathcal{G}_0]$. Then $\mathbb{E}[W \mid \mathcal{G}_0] = 0$ and, by linearity of conditional expectation,

$$\mathbb{E}[X \mid \mathcal{G}_1] - \mathbb{E}[X \mid \mathcal{G}_0] = \mathbb{E}\Big[X - \mathbb{E}[X \mid \mathcal{G}_0] \mid \mathcal{G}_1\Big] = \mathbb{E}[W \mid \mathcal{G}_1]. \tag{43}$$

Using equation 42, we can write

$$\begin{aligned}
\text{Var}\big(\mathbb{E}[X \mid \mathcal{G}_1] \mid \mathcal{G}_0\big) &= \mathbb{E}\Big[\big(\mathbb{E}[X \mid \mathcal{G}_1] - \mathbb{E}[\mathbb{E}[X \mid \mathcal{G}_1] \mid \mathcal{G}_0]\big)^2 \,\Big|\, \mathcal{G}_0\Big] \\
&= \mathbb{E}\Big[\big(\mathbb{E}[X \mid \mathcal{G}_1] - \mathbb{E}[X \mid \mathcal{G}_0]\big)^2 \,\Big|\, \mathcal{G}_0\Big] \\
&= \mathbb{E}\Big[\big(\mathbb{E}[W \mid \mathcal{G}_1]\big)^2 \,\Big|\, \mathcal{G}_0\Big].
\end{aligned} \tag{44}$$

By the conditional Jensen inequality for the convex function $\phi(u) = u^2$,

$$\big(\mathbb{E}[W \mid \mathcal{G}_1]\big)^2 \;\leq\; \mathbb{E}[W^2 \mid \mathcal{G}_1] \qquad \text{a.s.} \tag{45}$$

Taking $\mathbb{E}[\cdot \mid \mathcal{G}_0]$ on both sides and applying the tower property yields

$$\begin{aligned}
\text{Var}\big(\mathbb{E}[X \mid \mathcal{G}_1] \mid \mathcal{G}_0\big) &\leq \mathbb{E}\big[\mathbb{E}[W^2 \mid \mathcal{G}_1] \mid \mathcal{G}_0\big] = \mathbb{E}[W^2 \mid \mathcal{G}_0] \\
&= \mathbb{E}\big[(X - \mathbb{E}[X \mid \mathcal{G}_0])^2 \mid \mathcal{G}_0\big] = \text{Var}(X \mid \mathcal{G}_0).
\end{aligned} \tag{46}$$

This proves the variance inequality in equation 41.

Equality case. Equality in equation 45 holds iff W is $\mathcal{G}_1$-measurable, equivalently iff $\mathrm{Var}(W \mid \mathcal{G}_1) = 0$ a.s. Since $\mathbb{E}[X \mid \mathcal{G}_0]$ is $\mathcal{G}_0$-measurable and $\mathcal{G}_0 \subseteq \mathcal{G}_1$, this is equivalent to $\mathrm{Var}(X \mid \mathcal{G}_1) = 0$ a.s.

□

Remark. Proposition D.1 is a variance reduction statement for the random *state estimator* (induced by sampling M and/or the destination) 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 surrogates.

E Image Generation Experiment Details

This section provides additional implementation details and experimental setup for the compositional text-to-image generation experiments presented in Section 5.2.

E.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 in a discrete latent space and generates images through a non-autoregressive denoising process. The model uses a transformer backbone and was trained on large-scale image-text pairs.

E.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.

E.3 Reward Models

Our reward function 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. The reward is computed from the model's logits indicating prompt-image correspondence.

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 aesthetic quality of generated images.

Combined Reward. The final reward is a weighted combination:

$$R = R_{\text{fidelity}} + 0.5 \cdot R_{\text{aesthetic}} \tag{47}$$

This weighting was chosen to prioritize compositional correctness while maintaining visual quality.

E.4 Baseline Methods

We compare DDNO against the following inference-time baselines:

Base Model (FUDOKI). Standard sampling from the pre-trained FUDOKI model using the default sampling configuration. This establishes the baseline generation quality without any reward-based steering.

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

Reward Guidance. We implement discrete reward guidance (Nisonoff et al., 2025) using the same composite reward function as DDNO. We sweep guidance strength over $\lambda \in \{10, 100, 150, 300, 500, 1000\}$ and report the best-performing configuration.

Best-of-N Sampling. For each prompt, we generate N independent samples from the base model and select the sample with the highest reward score. We evaluate at $N = 50$ to match the computational budget of DDNO (50 optimization iterations). We use 50 NFEs for Best-of-N while we use 16 NFEs during noise optimization with DDNO. For the compute-matched comparison, we alter the amount of optimization steps for DDNO and the N generated images for Best-of-N.

Differentiable Surrogate (Unoptimized). The differentiable surrogate process described in Section 3 without any optimization of the initial noise logits. This ablation quantifies the fidelity gap introduced by the surrogate approximation.

E.5 DDNO Hyperparameters

Table 7 summarizes the hyperparameters used for DDNO on the GenEval benchmark.

Table 7: **DDNO hyperparameters for image generation.** Configuration used for GenEval experiments.

Hyperparameter	Value
Optimization iterations	50 (or compute-matched)
Learning rate	0.5
Optimizer	Adam
Surrogate temperature	1.0
Initial logit scale	1.0
Gradient clip norm	1.0
KL regularization weight	100.0

Denoising Configuration. We use the default FUDOKI sampling configuration with 50 denoising steps. The surrogate process maintains the same number of steps to ensure fair comparison with the base model.

Optimization Details. Optimization of the initial noise logits is performed using the Adam optimizer (Kingma, 2014). We employ gradient checkpointing to reduce memory consumption during backpropagation through the full denoising trajectory. All experiments are conducted in mixed precision (FP16) for computational efficiency.

E.6 Evaluation Protocol

Sample Generation. For each prompt in the GenEval benchmark, we run DDNO optimization starting from freshly initialized noise logits. The best sample across all optimization iterations (as measured by the reward function) is selected as the final output.

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

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.

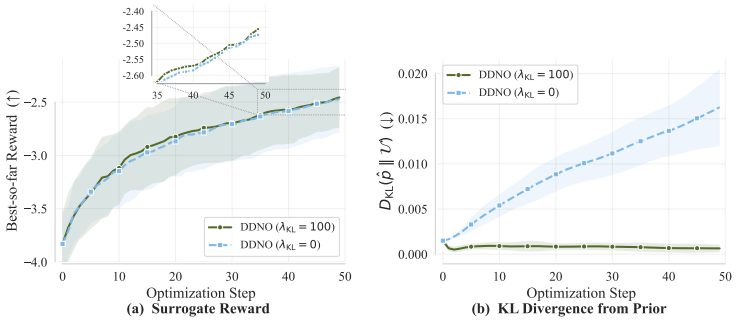

Figure 7: Ablation of KL regularization in DDNO. **(a)** Best-so-far surrogate reward over optimization steps. Both variants improve steadily; the zoomed inset reveals that $\lambda_{\mathrm{KL}} = 100$ attains marginally higher reward. **(b)** KL divergence from the prior. Without regularization ($\lambda_{\mathrm{KL}}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

E.7 Limitations and Failure Modes

We observe several limitations of DDNO in the image generation setting:

Surrogate Fidelity Gap. As shown in Table 1, the differentiable surrogate incurs a fidelity drop compared to standard discrete sampling (0.67 vs. 0.76 on GenEval). While DDNO optimization more than compensates for this gap, the initial fidelity loss represents an inherent trade-off of our approach.

Optimization Cost. DDNO requires multiple forward and backward passes through the diffusion model, making it more computationally expensive than single-sample generation. However, the amount of images is generated is the same as Best-of-N sampling while achieving superior results. We believe DDNO can be especially useful when it's crucial to reach very high rewards, that the base model is very unlikely to generate.

Reward Model Dependence. The quality of DDNO's outputs depends on the quality of the reward model. Imperfect reward models may lead to reward hacking, where generated images achieve high reward scores without genuinely satisfying the compositional requirements. The gap between reward model scores and GenEval automatic evaluation provides some indication of this phenomenon.

Complex Compositions. While DDNO substantially improves compositional generation, extremely complex prompts with many objects, attributes, and spatial relationships remain challenging. Performance degrades gracefully as compositional complexity increases.

F Topic Steering Experiment Details

This section provides implementation details and experimental setup for the topic steering experiments presented in Section 5.3.

F.1 Model and Architecture

We use GIDD-Unif-3B (von Rütte et al., 2025), a 3 billion parameter 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 based on the shared hard surrogate implementation.

F.2 Task Configuration

We steer generation toward specific semantic topics (science, sports) using natural language inference (NLI) as the reward signal. The reward is computed as the entailment probability between the generated text and a hypothesis of the form "This text is about {topic}." We evaluate across 10 diverse narrative prompts, designed to provide neutral starting points that do not bias the model toward any particular topic. The full list is provided in Table 8.

Table 8: **Topic steering prompts.** 10 narrative prompts used in the topic steering experiments. Each prompt is evaluated on both science and sports targets.

#	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.

F.3 Reward and Evaluation Models

Reward Model (Optimization). We use DeBERTa-v3-Large fine-tuned on MNLI, FEVER, ANLI, LingNLI, and WANLI (**??**) (`MoritzLaurer/DeBERTa-v3-large-mnli-fever-anli-ling-wanli`). The reward is the softmax probability of the "entailment" class when classifying the generated text against 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. This provides an independent assessment of topic adherence that is not directly optimized during DDNO.

Perplexity Model. We measure generation fluency using GPT-2 XL (Radford et al., 2019) perplexity, computed as the exponentiated cross-entropy loss over the generated token sequence.

F.4 Baseline: Best-of-N Sampling

For each prompt and topic, we generate N independent samples from the GIDD base model using the same denoising configuration (128 steps, cosine schedule, adaptive sampling at temperature 0.85) and select the sample with the highest reward. We evaluate at $N \in \{1, 30, 75, 150, 225, 300\}$ to characterize scaling behavior. For each value of N, we repeat the procedure across 10 evaluation samples per prompt.

F.5 DDNO HYPERPARAMETERS

Table 9 summarizes the DDNO hyperparameters used for topic steering.

Table 9: **DDNO hyperparameters for topic steering.** Configuration used across all 10 prompts for both science and sports targets.

HYPERPARAMETER	VALUE
OPTIMIZATION ITERATIONS (T)	0 / 10 / 25 / 50 / 75
LEARNING RATE	0.05
GUIDANCE SCALE (β)	10 000
KL WEIGHT (λ_{KL})	0.1
SURROGATE TEMPERATURE	0.85
INITIAL LOGIT SCALE	0.05
GRADIENT CLIP NORM	0.001
SEQUENCE LENGTH (SUFFIX)	64
DENOISING STEPS	128
EVAL SAMPLES PER PROMPT	10

Optimization Details. We optimize the initial noise logits $z \in \mathbb{R}^{L\times V}$ (where $L = 64$ and $V = 65{,}024$) using Adam (Kingma, 2014) with a learning rate of 0.05. The logits are initialized from a Gaussian distribution with standard deviation 0.05. Each optimization step backpropagates through the full 128-step denoising loop using Gumbel-STE with the shared hard surrogate implementation. Gradient checkpointing is enabled to reduce memory usage. The loss is

$$\mathcal{L} = -\beta \cdot R(x) + \lambda_{\text{KL}} \cdot D_{\text{KL}}(\text{softmax}(z) \,\|\, \mathcal{U}) \,, \tag{48}$$

where $R(x)$ is the NLI entailment probability for the target topic, $\beta = 10{,}000$ is the guidance scale, and $\lambda_{\text{KL}} = 0.1$ regularizes toward the uniform prior $\mathcal{U}$.

F.6 EVALUATION PROTOCOL

Sample Generation. For each prompt, topic, and condition, we produce 10 independent 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 generation selected from a group of N base-model samples.

Metrics. We report the following metrics averaged across prompts:

- **Reward** (R): mean NLI entailment probability from the optimization reward model (DeBERTa-v3-Large).
- **Evaluator** (E): mean entailment probability from the held-out evaluator (RoBERTa-Large-MNLI).
- **PPL**: GPT-2 XL perplexity of the final generated text.

Results are summarized in Figure 4.

G REASONING EXPERIMENT DETAILS

This section provides implementation details and experimental setup for the reasoning experiments presented in Section 5.3.

G.1 MODEL AND ARCHITECTURE

We use GIDD-Unif-3B (von Rütte et al., 2025), a 3 billion parameter 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 based on the shared hard surrogate implementation.

G.2 Task Configuration

We design 20 open-ended reasoning prompts spanning two difficulty levels.

Standard prompts (10). These cover everyday reasoning tasks such as giving advice on exam stress, explaining how to budget monthly expenses, handling a workplace disagreement, or comparing renewable and non-renewable energy sources.

Difficult prompts (10). These require more nuanced multi-step reasoning, including explaining the resource curse with three mechanisms, comparing climate–agriculture interactions between Egypt and Brazil, or explaining why Venus is hotter than Mercury despite being farther from the Sun.

The full list of prompts is provided in Table 10.

Table 10: **Reasoning prompts.** 20 open-ended prompts used in the reasoning experiments, split into standard and difficult categories.

#	Prompt
Standard (10 prompts)	
1	A friend feels stressed before exams. Provide practical ways to stay calm and focused.
2	Explain how to budget monthly expenses for someone who just started their first job.
3	A colleague disagrees with your project approach during a team meeting. How do you handle the situation constructively?
4	Describe the key differences between renewable and non-renewable energy sources and their environmental impact.
5	Someone asks you to explain how vaccines work to a skeptical family member. What would you say?
6	Your younger sibling wants to learn programming but feels overwhelmed. How would you guide them to get started?
7	A neighbor complains about noise from your apartment. Draft a thoughtful response that addresses their concern.
8	Explain the pros and cons of remote work versus office work for a company considering a permanent policy change.
9	A student asks why learning history is important when they want to pursue a career in technology. How do you respond?
10	Describe how someone can develop better critical thinking skills in their daily life.
Difficult (10 prompts)	
11	Why can a country be rich in natural resources and still remain poor? Explain with three mechanisms.
12	Why does increasing atmospheric CO_2 warm the Earth? Give a short explanation without equations.
13	Explain why Venus is hotter than Mercury, even though Mercury is closer to the Sun.
14	Why can antibiotics treat bacterial infections but not viral infections? Explain for a non-expert.
15	Why do some countries have many natural harbors while others have long coastlines with few major ports?
16	Compare why Singapore and Switzerland both became major economic hubs despite very different geography.
17	Compare how climate affects agriculture in Egypt and Brazil, and explain how that shapes their economies.
18	Why can access to the sea matter for economic development? Compare one landlocked country and one coastal country.
19	Explain why drought can increase both food prices and political instability.
20	A student asks why some diseases spread faster in cities than in rural areas. Explain using population density, mobility, and public health capacity.

G.3 Reward Model

QRM (Optimization and Evaluation). We use QRM-Llama3.1-8B-v2 (Dorka, 2024), an 8 billion parameter reward model based on Llama 3.1 that predicts the full reward distribution via quantile regression over 19 quantiles ($\tau = 0.05, 0.10, \ldots, 0.95$). A learned gating network weights attributes conditioned on the prompt. The scalar reward signal is the expected value (mean over predicted quantiles).

Prompt Format. Generated text is formatted as a chat turn using the Llama 3.1 chat template before being scored by QRM: `[BOS] user: {prompt} assistant: {response} [EOS].`

Perplexity Model. We measure generation fluency using GPT-2 XL (Radford et al., 2019) perplexity, computed as the exponentiated cross-entropy loss over the generated token sequence.

G.4 Baseline: Best-of-N Sampling

For each prompt, we generate $N = 45$ independent samples from the GIDD base model using the same denoising configuration (128 steps, cosine schedule, adaptive sampling at temperature 0.85) and select the sample with the highest QRM reward. This is repeated for 5 independent evaluation groups, yielding 225 total base-model generations per prompt, from which 5 selected samples (one per group) are retained.

G.5 DDNO Hyperparameters

Table 11 summarizes the DDNO hyperparameters used for the reasoning experiments.

Table 11: **DDNO hyperparameters for reasoning.** Configuration used across all 20 reasoning prompts.

Hyperparameter	Value
Optimization iterations (T)	15 / 25
Learning rate	0.3
Guidance scale (β)	10 000
KL weight (λ_{KL})	0.1
Surrogate temperature	0.85
Initial logit scale	0.05
Gradient clip norm	0.0001
Sequence length (suffix)	64
Denoising steps	128
Eval samples per prompt	5
BoN samples (N)	45

Optimization Details. We optimize the initial noise logits $z \in \mathbb{R}^{L \times V}$ (where $L = 64$ and $V = 28{,}416$) using Adam (Kingma, 2014) with a learning rate of 0.3. The logits are initialized from a Gaussian distribution with standard deviation 0.05 and clamped to $[-50, 50]$ for numerical stability. Each optimization step backpropagates through the full 128-step denoising loop using Gumbel-STE with deterministic per-step Gumbel noise. Gradient checkpointing is enabled to reduce memory usage. The loss is

$$\mathcal{L} = -\beta \cdot R(x) + \lambda_{\text{KL}} \cdot D_{\text{KL}}(\text{softmax}(z) \,\|\, \mathcal{U}) \,, \tag{49}$$

where $R(x)$ is the QRM reward, $\beta = 10{,}000$ is the guidance scale, and $\lambda_{\text{KL}} = 0.1$ regularizes toward the uniform prior $\mathcal{U}$.

Two optimization budgets. We evaluate DDNO at $T = 15$ and $T = 25$ optimization iterations to study the effect of longer optimization on reward and text quality. Both use identical hyperparameters otherwise.

G.6 Evaluation Protocol

Sample Generation. For each prompt and condition, we produce 5 independent samples. For DDNO, each sample is an independent optimization run from freshly initialized noise logits. For Best-of-N, each sample is the highest-QRM-reward generation selected from a group of 45 base-model samples.

Metrics. We report the following metrics averaged across prompts within each difficulty category:

- **QRM Reward**: the best reward achieved along the optimization trajectory (for DDNO) or the reward of the selected sample (for BoN).
- **QRM Init**: the mean QRM reward before optimization or selection—averaged over all 225 base-model candidates for BoN, or over the 5 initial surrogate outputs for DDNO.
- **PPL**: GPT-2 XL perplexity of the final generated text.
- **PPL Init**: perplexity of the initial generation before optimization or selection.
- **Entropy**: token-level entropy of the generated sequences.

Results are summarized in Table 3.