PTERA: PROBABILISTIC TEXT ENCODER FOR REWARD ALIGNMENT IN DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Fine-tuning text-to-image models toward human preference rewards has become a common approach for improving text alignment. Most existing methods, however, focus on updating the diffusion model parameters, while reward-based adaptation of the text encoder remains relatively underexplored despite promising reward gain and training efficiency. Moreover, existing text encoder fine-tuning methods typically optimize a single deterministic embedding for each prompt, which can be susceptible to reward over-optimization and provides no training mechanism for exploring alternative conditioning vectors. In this work, we propose **P**robabilistic **T**ext **E**ncoder for **R**eward **A**lignment (**PTERA**), a simple yet effective text encoder fine-tuning framework based on direct reward backpropagation. PTERA replaces deterministic conditioning with a trainable probability distribution over text embeddings and derives a tractable KL-regularized objective for learning this distribution while regularizing the divergence from a stochastic reference constructed around the pre-trained text embeddings. Compared with previous text encoder adaptation, PTERA exhibits more robust reward alignment. We further introduce **PTERA-TTA**, which exploits the text embedding distribution to perform trajectory-dependent test-time alignment.



Figure 1: Representative results of PTERA. (**Left.**) Generated images for the prompt *"New York Skyline with 'Diffusion' written with fireworks on the sky."* (**Right.**) PickScore comparison across SD1.5 and SDXL. PTERA consistently improves both backbones. Notably, SD1.5+PTERA outperforms vanilla SDXL.

1 INTRODUCTION

Text-to-image (T2I) diffusion models have achieved remarkable progress in generating high-quality images from natural language prompts. However, pre-trained models do not necessarily satisfy the preferences required by a specific downstream application, such as stronger text alignment (Xu et al., 2023), safe generation (Gandikota et al., 2023; Lyu et al., 2024), or personalized visual preferences (Gal et al., 2023; Ruiz et al., 2023). A common approach is therefore to align the pre-trained model toward a human preference reward, either by fine-tuning its parameters (Black et al., 2024; Clark et al., 2024; Prabhudesai et al., 2023; Domingo i Enrich et al., 2025) or by steering individual generations at test time (Holderrieth et al., 2026; Potaptchik et al., 2025; Kim et al., 2026b).

While most alignment methods operate on the diffusion backbone, text conditioning provides a substantially smaller and computationally attractive adaptation space (Ye et al., 2025; Lee et al., 2026). Prior works have shown that fine-tuning the text encoder can effectively optimize generation quality (Li et al., 2024; Chen et al., 2024), and that perturbing or optimizing text embeddings at inference time can steer individual generations (Sadat et al., 2024; Na et al., 2025; Hu et al., 2026). Nevertheless, these two directions have largely been studied separately. Text encoder fine-tuning

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

learns a single deterministic embedding for each prompt, whereas embedding space exploration searches nearby conditioning vectors using a manually specified exploration mechanism.

In this work, we unify these two perspectives by modeling the text conditioning itself as a probability distribution. Probabilistic conditioning has previously been explored in generative and representation models as a mechanism for capturing multiple plausible representations (Zhang et al., 2017; Chun, 2024; Zhao et al., 2025). Building on this idea, we replace the deterministic text embedding of a pre-trained diffusion model with a local Gaussian distribution and make this distribution trainable under a human preference reward. This formulation allows the encoder to learn not only where an optimized text embedding should lie, but also how the embedding space should be explored.

We propose **P**robabilistic **T**ext **E**ncoder for **R**eward **A**lignment (**PTERA**). PTERA attaches a lightweight residual module to a frozen pre-trained text encoder and parameterizes a Gaussian distribution through its mean and coordinate-wise scale. Since marginalizing this latent text embedding turns the generated image distribution into a mixture, the standard KL-regularized reward alignment objective becomes difficult to evaluate directly. We derive a tractable training framework based on upper bounds of the resulting mixture KL divergence, enabling PTERA to learn a reward-aware text embedding distribution while preserving the behavior of the pre-trained text-to-image model.

Importantly, the learned distribution can also be exploited at inference time. We introduce **PTERA-TTA**, which combines PTERA with rollout test-time alignment (rollout TTA) by sampling candidate text embeddings from the learned distribution and selecting candidates that improve the reward for individual denoising trajectories. PTERA therefore connects amortized text encoder optimization with instance-specific text embedding exploration within a single probabilistic framework. As illustrated in Figure 1, PTERA yields strong reward improvements across both SD1.5 and SDXL, with PTERA-TTA providing further gains through test-time exploration.

2 PRELIMINARIES

2.1 DIFFUSION MODELS

Diffusion models generate a clean sample x_0 from an initial noise $x_T \sim p_T$ via iterative denoising steps, where x_t denotes the intermediate state at timestep t (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021b). In text-to-image (T2I) generation, the denoising process is conditioned on a text prompt c to generate an image aligned with its textual description. Let E_φ denote a text encoder parameterized by φ, which maps c to a text embedding $E_\varphi(c)$, and let θ denote the parameters of the diffusion model. The text encoder E_φ is generally adopted from a pre-trained model, such as CLIP (Radford et al., 2021) or T5 (Raffel et al., 2020).

Given the text embedding $E_\varphi(c)$, the reverse generative process is characterized by

$$p_{\theta,\varphi}(x_{0:T}|c) \coloneqq p_T(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t, E_\varphi(c)). \tag{1}$$

The resulting distribution of the generated image is denoted by $p_{\theta,\varphi}(x_0|c) \coloneqq p_\theta(x_0|E_\varphi(c))$. Diffusion models are commonly trained by maximizing a variational lower bound on the data log-likelihood and admit several parameterizations, including $\hat{x}_0$, noise, score, and velocity prediction (Ho et al., 2020; Song et al., 2021b; Karras et al., 2022; Esser et al., 2024). Throughout this work, we use the $\hat{x}_0$ prediction approach and denote the prediction of x_0 from an intermediate state x_t as follows: $\hat{x}_{0|t}^{\theta,\varphi} \coloneqq \hat{x}_{0|t}^{\theta}(x_t, E_\varphi(c))$ or simply $\hat{x}_{0|t}$.

2.2 REWARD MAXIMIZATION WITH REVERSE KL REGULARIZATION

Despite notable advances in conditional image generation, adapting diffusion models to specific downstream objectives such as text alignment remains an active area of research. These objectives can typically be formulated as maximizing a human-preferred reward function $\mathcal{R} : \mathcal{X} \times \mathcal{C} \to \mathbb{R}$, where $\mathcal{X}$ and $\mathcal{C}$ denote the image and prompt spaces, respectively. Given a conditional image distribution $p_{\theta,\varphi}(x_0|c)$, parameterized by diffusion model parameter θ and text encoder parameter φ, the corresponding reward-maximization objective is

$$\max_{\theta \text{ or } \varphi} \mathbb{E}_{c,x_0 \sim p_{\theta,\varphi}}[\mathcal{R}(x_0, c)]. \tag{2}$$

When the reward function is differentiable, this objective can be optimized by directly backpropagating the reward gradient through the denoising process with respect to either θ or φ (Clark et al., 2024; Prabhudesai et al., 2023; Xu et al., 2023; Li et al., 2024). Alternatively, the reverse diffusion process can be formulated as a Markov decision process, enabling reinforcement-learning-based optimization of the same expected reward (Black et al., 2024; Chen et al., 2024).

Optimizing Equation 2 alone, however, is susceptible to reward over-optimization, whereby the fine-tuned model exploits imperfections in the reward model while substantially deviating from the pre-trained model distribution. A standard remedy is to regularize the optimized distribution against a fixed reference model using the reverse KL regularization term. This yields the following objective (Christiano et al., 2017; Ouyang et al., 2022), which has been widely adopted for reward-based diffusion model fine-tuning (Fan et al., 2023; Uehara et al., 2024; Domingo i Enrich et al., 2025; Liu et al., 2025):

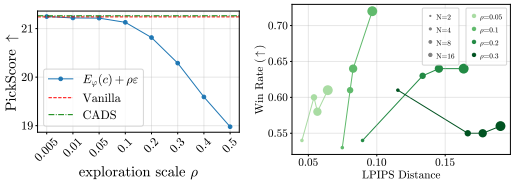
$$\max_{\theta \text{ or } \varphi} \mathbb{E}_{x_0 \sim p_{\theta,\varphi}}[\mathcal{R}(x_0, c)] - \alpha D_{\mathrm{KL}}(p_{\theta,\varphi}(x_0|c) \,\|\, p_{\theta_{\mathrm{ref}},\varphi_{\mathrm{ref}}}(x_0|c)) \tag{3}$$

where $\alpha > 0$ controls the trade-off between reward maximization and preservation of the reference distribution. Directly evaluating the KL divergence between the marginal image distributions is generally intractable. Prior work (Fan et al., 2023) addresses this issue by upper-bounding the marginal divergence with a tractable trajectory-level regularizer over the reverse diffusion process.

2.3 Rollout Test-time Alignment

Another approach to reward maximization in diffusion models is test-time alignment (TTA), which steers the generation process without fine-tuning. These methods eliminate training costs at the expense of modest additional inference-time computation, and they can often be orthogonally combined with fine-tuning methods to further improve reward alignment. Specifically, TTA steers the sampling process toward the optimal target distribution p_{tar} induced by the objective in Equation 3, given by:

$$p_{\mathrm{tar}}(x_0|c) \propto p_{\theta_{\mathrm{ref}},\varphi_{\mathrm{ref}}}(x_0|c) \exp\left(\mathcal{R}(x_0, c)/\alpha\right). \tag{4}$$



(a) Effect of isotropic embedding exploration. (b) Rollout TTA via text embedding exploration.

Figure 2: Empirical analysis of text embedding exploration on SD1.5. **(a)** PickScore (Kirstain et al., 2023) of the generated images under varying exploration scales ρ. **(b)** Win rate and LPIPS distance relative to the corresponding vanilla sample using Algorithm 1. Larger markers indicate a greater number of candidates N, while darker shades indicate a larger ρ.

Recent work (Kim et al., 2026b) shows that rollout-based TTA (Rollout TTA) methods are less prone to reward hacking and particle collapse than gradient guidance (Bansal et al., 2024; Na et al., 2025; Kim et al., 2026a) and sequential Monte Carlo methods (Singhal et al., 2025; Li et al., 2025). Rollout TTA (Holderrieth et al., 2026; Potaptchik et al., 2026; 2025) replaces the standard x_0 prediction $\hat{x}_{0|t}$ with a reward-weighted estimate $\hat{x}^{\mathrm{TTA}}_{0|t}$ that approximates Equation 4:

$$\hat{x}^{\mathrm{TTA}}_{0|t} := \frac{\sum_i^N \tilde{x}^i_{0|t} \exp(\mathcal{R}(\tilde{x}^i_{0|t}, c)/\alpha)}{\sum_j^N \exp(\mathcal{R}(\tilde{x}^j_{0|t}, c)/\alpha)}, \qquad \tilde{x}^i_{0|t} = \textbf{Candidate}(x_t, i), \tag{5}$$

where the **Candidate** function proposes an x_0 prediction given x_t. A key challenge in Rollout TTA is generating diverse $\tilde{x}^i_{0|t}$ proposals given x_t. Thus, the deterministic one-step prediction $\hat{x}_{0|t}$ cannot be directly used. Prior works introduce stochasticity through inner flows (Holderrieth et al., 2026; Potaptchik et al., 2025), or forward noise injection (Kim et al., 2026b) in the sample space; however, these methods perturb the image latent, which can disrupt the spatial information preserved in x_t. Recent studies (Sadat et al., 2024; Hu et al., 2026) have shown that randomly perturbing the text embeddings provides an effective space for improving diffusion models (See Proposition 1 for our theoretical explanation of random perturbation). Motivated by this, we introduce a more principled way to inject stochasticity in the text embedding space and explore its use for Rollout TTA.

Text embedding exploration for Rollout TTA. Figure 2a empirically examines isotropic text embedding exploration, $z = E_\varphi(c) + \rho\varepsilon$, where $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, on SD1.5 with CADS (Sadat et al., 2024),

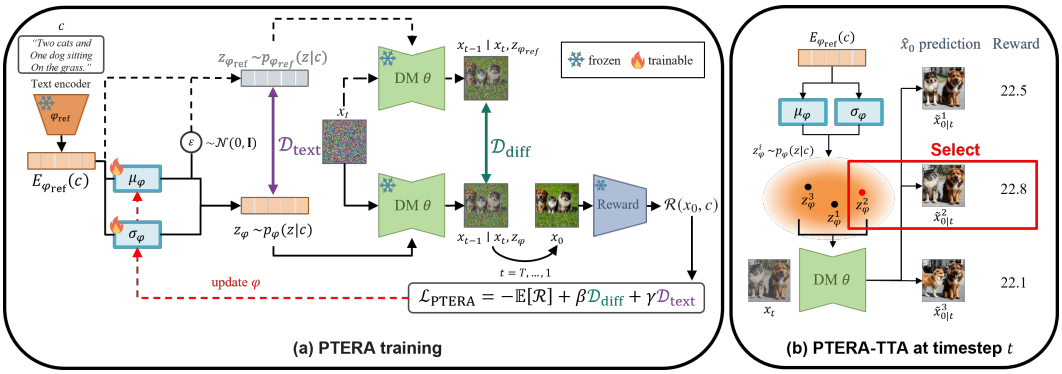


Figure 3: **(a)** Overview of PTERA training and **(b)** PTERA-TTA, which exploits the PTERA distribution.

a training-free diffusion sampling method that also injects Gaussian noise into the text condition. For small exploration scales, particularly $\rho \leq 0.1$, PickScore remains close to that of the unperturbed model, suggesting that local perturbations can introduce conditioning diversity without substantially degrading the reference generation quality. Consistent with this observation, Proposition 1 bounds the induced change in the output distribution under local embedding perturbations.

We next instantiate the **Candidate** function in Equation 5 by predicting x_0 under independently perturbed text embeddings and obtain the greedy selection procedure in Algorithm 1. As shown in Figure 2b, this simple local search achieves a win rate above 0.5 against the corresponding vanilla samples across the tested configurations. These observations motivate using a local Gaussian distribution around the pre-trained text embedding as the reference exploration distribution in PTERA.

3 Method

PTERA learns a prompt-conditioned distribution over text embeddings while keeping the pre-trained diffusion backbone frozen. In contrast to deterministic text encoder alignment, which maps each prompt to a single conditioning embedding, PTERA learns a reward-aware distribution that can support both reward-based fine-tuning and subsequent test-time exploration. Before introducing this probabilistic formulation, we first define its deterministic counterpart.

DTERA: Deterministic text encoder alignment. TextCraftor (Li et al., 2024) optimizes a deterministic text encoder by directly backpropagating a reward through the denoising process. We consider its KL-regularized variant by optimizing Equation 3 with respect to the text encoder parameter φ while keeping the diffusion backbone fixed. Following Fan et al. (2023), we replace the marginal KL term with its trajectory-level upper bound. We refer to this deterministic formulation as **DTERA**. PTERA extends this formulation to a trainable distribution over text embeddings.

3.1 PTERA: Probabilistic Text Encoder for Reward Alignment

We now extend the reverse KL-regularized reward objective in Equation 3 from deterministic text conditioning to a trainable distribution over text embeddings.

Text embedding as a stochastic latent variable. For a prompt c, we extend the deterministic conditioning embedding to a stochastic latent variable z and let $\Pi_c = p_\varphi(\cdot \mid c)$ denote its trainable mixing distribution. Throughout this section, z denotes a stochastic text embedding, whereas $E_{\varphi_{\text{ref}}}(c)$ denotes the deterministic embedding produced by the pre-trained text encoder.

Because the reverse KL objective compares the learned model against a fixed reference model, introducing probabilistic target conditioning also requires a reference mixing measure in the same text embedding space. A deterministic reference encoder can formally be represented by the point mass $\delta_{E_{\varphi_{\text{ref}}}(c)}$. However, such a degenerate reference does not provide a finite encoder KL regularizer for a non-degenerate target distribution. We therefore use the local Gaussian $\Pi_{c^{\text{ref}}} = \mathcal{N}\left(E_{\varphi_{\text{ref}}}(c), \rho^2 \mathbf{I}\right)$

as a stochastic surrogate for the deterministic reference conditioning. This choice is motivated by the local embedding exploration in Figure 2 and the stability result in Proposition 1.

Once the text embedding becomes a latent variable, the target and reference distributions marginalize over different mixing measures. As a result, the KL divergence no longer decomposes directly into point-wise conditional KL terms. To address this problem, Theorem 1 presents a coupling-based KL upper bound for mixture distributions (Nguyen, 2013) reformulated for continuous text embedding spaces.

Theorem 1 (Upper bound of the KL regularization)**.** *Let $p(x \mid z)$ be a generative kernel from the text embedding space to the sample space. For any probability measure Π on the text embedding space, define the induced mixture distribution as*

$$P_{\Pi} \coloneqq \int p(x \mid z)\, \Pi(dz). \tag{6}$$

Conditioned on a text prompt c, let Π_c and $\Pi_{c^{\mathrm{ref}}}$ denote the target and reference mixing measures, respectively. When $x = x_0$, the divergence $D_{\mathrm{KL}}\big(P_{\Pi_c} \,\big\|\, P_{\Pi_c^{\mathrm{ref}}}\big)$ is the probabilistic-conditioning counterpart of the reverse KL term in Equation 3, now between the output distributions induced by the target and reference text embedding mixing measures. Then

$$D_{\mathrm{KL}}\big(P_{\Pi_c} \,\big\|\, P_{\Pi_{c^{\mathrm{ref}}}}\big) \;\le\; \inf_{q \in \mathcal{Q}(\Pi_c, \Pi_{c^{\mathrm{ref}}})} \int D_{\mathrm{KL}}\big(p(x \mid z) \,\big\|\, p(x \mid z^{\mathrm{ref}})\big)\; q(dz, dz^{\mathrm{ref}}), \tag{7}$$

whenever the right-hand side is finite, where $\mathcal{Q}(\Pi_c, \Pi_{c^{\mathrm{ref}}})$ denotes the set of couplings on the product of the text embedding space with marginals Π_c and $\Pi_{c^{\mathrm{ref}}}$.

The proof is provided in Appendix A.3. Although Theorem 1 is stated for a generic sample x, it applies to any choice of sample space. In particular, taking $x = x_{0:T}$ yields the corresponding bound over full diffusion trajectories, which is the instantiation used below to define $\mathcal{D}_{\mathrm{diff}}$. Theorem 1 replaces the intractable KL divergence between two mixtures with a comparison between their conditional distributions under a coupling of the corresponding mixing measures. For any $q \in \mathcal{Q}(\Pi_c, \Pi_{c^{\mathrm{ref}}})$, the expected conditional KL provides an upper bound on the KL divergence between the sample distributions, while optimizing over q yields the tightest bound within this family. Thus, the remaining problem is to construct a tractable pairing between the target and reference text embedding distributions.

Tractable Gaussian Reparameterization. To parameterize the probabilistic encoder and construct tractable regularizers, we use a Gaussian reparameterization (Kingma & Welling, 2014; Chun, 2024). Given a prompt c, PTERA predicts a mean $\mu_\varphi(c)$ and a coordinate-wise scale $\sigma_\varphi(c)$, defining

$$p_\varphi(z \mid c) = \mathcal{N}\left(z; \mu_\varphi(c), \mathrm{diag}\big(\sigma_\varphi^2(c)\big)\right). \tag{8}$$

Samples from this distribution are obtained through

$$z_\varphi(c, \varepsilon) = \mu_\varphi(c) + \sigma_\varphi(c) \odot \varepsilon, \quad \varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{9}$$

The reference mixing distribution is recovered by setting $\mu_{\varphi_{\mathrm{ref}}}(c) = E_{\varphi_{\mathrm{ref}}}(c)$ and $\sigma_{\varphi_{\mathrm{ref}}}(c) = \rho\mathbf{1}$, corresponding to the local Gaussian reference distribution introduced above. In practice, μ_φ and σ_φ are implemented as residual heads on the pre-trained text encoder in Figure 3a.

The reparameterized construction suggests two complementary ways to compare the target and reference models. First, the two encoders can be evaluated using the *same noise* ε, producing the paired embeddings $z_\varphi(c, \varepsilon)$ and $z_{\varphi_{\mathrm{ref}}}(c, \varepsilon)$. This shared-$\varepsilon$ matching directly defines a coupling between the two mixing measures and measures how differently the frozen diffusion model responds to the paired embeddings, yielding the diffusion-level regularizer $\mathcal{D}_{\mathrm{diff}}$ in Figure 3.

Alternatively, the two models can be compared at a *shared latent embedding* z. Because they share the same conditional diffusion kernel $p_{\theta_{\mathrm{ref}}}(x_{0:T} \mid z)$, the joint KL reduces to the text embedding KL, which also controls the output-level divergence by data processing. This gives the text-level regularizer $\mathcal{D}_{\mathrm{text}}$ in Figure 3.

These two constructions provide complementary surrogates for the same output-level KL regularization. We combine them with nonnegative weights β and γ satisfying $\beta + \gamma = \alpha$, where α is the reverse KL coefficient in Equation 3. This preserves the overall regularization scale of the original objective while allowing the relative contribution of the diffusion-level and text-level regularizers to be adjusted. We formally define the two regularizers and the resulting PTERA objective below.

Definition 1 (PTERA surrogate loss)**.** Given the probabilistic encoders in Equation 8, consider the following two matching schemes between the target encoder φ and the reference encoder φ_{ref}:

(1) (shared-ε matching, $\mathcal{D}_{\mathrm{diff}}$) Pairing the two encoders through a shared noise $\varepsilon \sim \mathcal{N}(0, \mathbf{I})$ induces the coupling $\mathrm{Law}\big(z_\varphi(c,\varepsilon),\ z_{\varphi_{\mathrm{ref}}}(c,\varepsilon)\big) \in \mathcal{Q}(\Pi_c, \Pi_{c^{\mathrm{ref}}})$, and defines

$$\mathcal{D}_{\mathrm{diff}}(\varphi; c) \coloneqq \mathbb{E}_\varepsilon\Big[D_{\mathrm{KL}}\big(p_{\theta_{\mathrm{ref}}}\big(x_{0:T} \mid z_\varphi(c,\varepsilon)\big) \,\big\|\, p_{\theta_{\mathrm{ref}}}\big(x_{0:T} \mid z_{\varphi_{\mathrm{ref}}}(c,\varepsilon)\big)\big)\Big]. \tag{10}$$

(2) (shared-z matching, $\mathcal{D}_{\mathrm{text}}$) Pairing the two models through a shared latent z at the level of the joint distributions $p_\varphi(z \mid c)\, p_{\theta_{\mathrm{ref}}}(x_{0:T} \mid z)$ and $p_{\varphi_{\mathrm{ref}}}(z \mid c)\, p_{\theta_{\mathrm{ref}}}(x_{0:T} \mid z)$ defines

$$\mathcal{D}_{\mathrm{text}}(\varphi; c) \coloneqq D_{\mathrm{KL}}(p_\varphi(z \mid c) \,\|\, p_{\varphi_{\mathrm{ref}}}(z \mid c))\,. \tag{11}$$

Given a reward function $\mathcal{R}$, using the weights β, γ specified above, the *PTERA objective* is

$$\mathcal{L}_{\mathrm{PTERA}}(\varphi) \coloneqq \mathbb{E}_c\Big[\, -\mathbb{E}_{\substack{z\sim p_\varphi(\cdot|c)\\ x_0\sim p_\theta(\cdot|z)}}\big[\mathcal{R}(x_0,c)\big] \;+\; \beta\,\mathcal{D}_{\mathrm{diff}}(\varphi; c) \;+\; \gamma\,\mathcal{D}_{\mathrm{text}}(\varphi; c)\Big], \tag{12}$$

minimized over the encoder parameters φ while the diffusion model θ is kept frozen.

The fully tractable formulas for each regularizer are derived in Equations 40 and 45 of Appendix A.4. Furthermore, Proposition 2 in Appendix A.5 analyzes the resulting surrogate in an idealized linear Gaussian setting (Guo et al., 2024) and establishes that it is controlled between the true marginal KL divergence and a scaled version of it (See Proposition 2 in Appendix A.5). This result provides a formal justification for combining $\mathcal{D}_{\mathrm{diff}}$ and $\mathcal{D}_{\mathrm{text}}$, while allowing their relative strengths to be adjusted according to the desired balance between reward improvement and reference preservation. See Appendix D.2 for the experimental details and additional results with regard to the role of β, γ.

3.2 PTERA-TTA

We next examine how the reward component of Equation 12 shapes the learned text embedding distribution. Consider the reparameterization in Equation 9 and view the terminal reward as a differentiable function of z, suppressing its dependence on the generated trajectory and prompt for notational simplicity. Defining $g \coloneqq \partial\mathcal{R}/\partial z$, the reward gradients are

$$\frac{\partial \mathcal{R}}{\partial \mu_\varphi} = \frac{\partial \mathcal{R}}{\partial z}\frac{\partial z}{\partial \mu_\varphi} = g, \quad \frac{\partial \mathcal{R}}{\partial \sigma_\varphi} = \frac{\partial \mathcal{R}}{\partial z}\frac{\partial z}{\partial \sigma_\varphi} = g \odot \varepsilon. \tag{13}$$

Equation 13 provides a first-order interpretation of the distribution learned by PTERA. Under gradient ascent, μ_φ moves toward directions that locally increase the reward, while each stochastic update adjusts σ_φ according to whether the sampled perturbation is locally aligned with reward improvement. Together with the regularization terms in $\mathcal{L}_{\mathrm{PTERA}}$, these updates yield a distribution that adapts to the reward landscape while remaining close to the reference. We refer to the resulting conditional distribution as a *reward-aware text embedding distribution*.

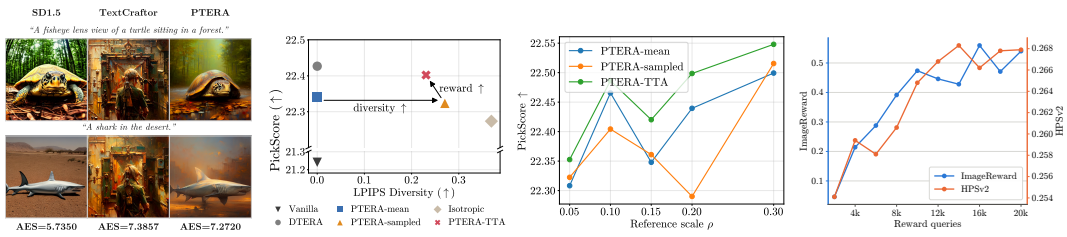
PTERA-TTA (Algorithm 1) exploits this reward-aware distribution as a proposal distribution for test-time exploration in the text embedding space. At a selected denoising timestep t, it keeps the diffusion state x_t fixed, samples candidate embeddings $\{z_\varphi^i\}_{i=1}^N \sim p_\varphi(z \mid c)$, and evaluates their rollout predictions using the reward, as illustrated in Figure 3b. The selected embedding is then used to continue the denoising trajectory. This shifts candidate generation from the diffusion state space to a text embedding space that has already been adapted by reward optimization, rather than exploring an isotropic neighborhood around a fixed text embedding. As shown in Appendix D.3, this text space exploration yields a more local reward-deviation trade-off than image-space rollout alternatives, while avoiding redundant diffusion evaluations under conditional generation.

4 Experiments

Experimental Details. We compare PTERA against a broad set of reward maximization methods operating at different levels of a T2I diffusion pipeline. DDPO (Black et al., 2024) optimizes the diffusion-model parameters θ through reinforcement learning, whereas TexForce (Chen et al.,

Table 1: Quantitative results on DrawBench and HPSv2. † denotes *reproduced* methods using the official implementations, whereas methods without † use the official checkpoints when available. The *Target* column specifies the reward used for training or TTA. For DTERA and PTERA, values in parentheses denote the regularization coefficient(s). For HPSv2, we additionally report HPSv2 scores by prompt categories. The best and second-best results in each column are shown in **bold** and underlined, respectively.

Dataset		DrawBench					HPSv2			
Method	**Target**	**PS**	**IR**	**HPSv2**	**CLIP**	**AES**	**Animation**	**Concept Art**	**Painting**	**Photo**
SD1.5	-	21.242	-0.039	0.246	26.940	5.735	0.269	0.221	0.226	0.249
TextCraftor	-	21.110	0.015	0.275	24.582	6.560	0.314	**0.314**	**0.302**	**0.287**
TexForce	-	21.246	-0.024	0.248	26.651	5.905	0.277	0.257	0.244	0.258
Promptist	-	20.934	-0.114	0.229	24.131	**6.605**	0.263	0.227	0.228	0.234
DDPO†	PS	21.254	0.043	0.243	26.806	5.751	0.265	0.231	0.230	0.248
TextCraftor†	PS	19.977	-1.308	0.202	20.792	5.045	0.214	0.217	0.201	0.200
TextCraftor†	IR	20.929	-0.140	0.253	25.364	5.579	0.277	0.250	0.239	0.266
TexForce†	PS	21.177	-0.045	0.243	26.097	5.795	0.277	0.237	0.237	0.257
TexForce†	IR	21.135	-0.075	0.247	26.297	5.860	0.268	0.239	0.234	0.254
AM†	PS	22.044	0.176	0.251	26.806	6.002	0.278	0.237	0.224	0.247
AM†	IR	21.417	0.346	0.253	**27.495**	5.896	0.286	0.238	0.229	0.253
DTERA (0.2)	PS	22.413	0.076	0.262	25.232	5.953	0.283	0.271	0.249	0.257
DTERA (0.4)	PS	22.047	-0.029	0.254	25.208	6.162	0.288	0.262	0.248	0.263
PTERA (0.1, 0.1)	PS	22.466	0.250	0.268	25.987	5.933	0.293	0.266	0.241	0.257
PTERA (0.1, 0.1)	IR	21.272	**0.557**	0.269	26.554	6.047	0.302	0.277	0.262	0.256
PTERA (0.1, 0.1)	HPSv2	21.307	0.217	**0.289**	26.153	5.645	**0.330**	0.307	0.286	0.283
PTERA-TTA (0.1, 0.1)	PS	**22.562**	0.327	0.269	26.074	5.918	0.296	0.269	0.246	0.256



(a) Reward over-optimization. (b) Reward-aware exploration. (c) Reference ρ sensitivity. (d) Reward dynamics.

Figure 4: Analysis of the characteristics of PTERA. **(a)** PTERA mitigates reward over-optimization compared with TextCraftor when optimizing AES. **(b)** Diversity induced by text embedding exploration of PTERA on DrawBench under the same initial noise and a deterministic DDIM (Song et al., 2021a) sampler. **(c)** Sensitivity to the reference scale ρ. **(d)** Validation performance of ImageReward steered PTERA over training.

2024) and TextCraftor (Li et al., 2024) optimize the text encoder parameters φ using reinforcement learning and direct reward backpropagation, respectively. These methods optimize the objective in Equation 2. Promptist (Hao et al., 2023), in contrast, performs optimization directly in the discrete prompt space. We further compare against methods based on the reverse KL-regularized objective in Equation 3, including Adjoint Matching (Domingo i Enrich et al., 2025) (AM), DTERA, and a KL-regularized variant of DRaFT (Clark et al., 2024), which we refer to as DRaFT+.

We use the prompts from DrawBench (Saharia et al., 2022) for fine-tuning and evaluate generalization on DrawBench as well as the held-out HPSv2 (Wu et al., 2023), PartiPrompts (Yu et al., 2022), and GenEval (Ghosh et al., 2023) benchmarks. We consider PickScore (Kirstain et al., 2023) (PS), ImageReward (Xu et al., 2023) (IR), and HPSv2 as steering rewards. In addition to the reward used for optimization, we report the remaining preference rewards as held-out metrics, together with CLIP Score (CLIP), Aesthetic Score (Schuhmann, 2022) (AES), and the GenEval score. We standardize evaluation to 50-step DDIM, while retaining method-specific training recipes unless otherwise stated. Further implementation details are provided in Appendix C.

T2I Generation. We first conduct the primary T2I generation experiments using SD1.5. All models are fine-tuned using only the prompt set from DrawBench. To parameterize the latent text embedding distribution, we augment the pre-trained text encoder with a lightweight residual module based on multi-head attention (Cordonnier et al., 2020). The module contains separate mean and scale heads, μ_φ and σ_φ. The detailed architecture is reported in Table 5. As shown in Table 1 and Figure 6a, PTERA achieves consistent improvements compared with existing fine-tuning methods. In particular, compared with its deterministic counterpart DTERA, PTERA exhibits stronger reward generalization on several held-out preference metrics, indicating that probabilistic text embedding modeling provides benefits beyond optimizing a single deterministic embedding. Furthermore, Fig-

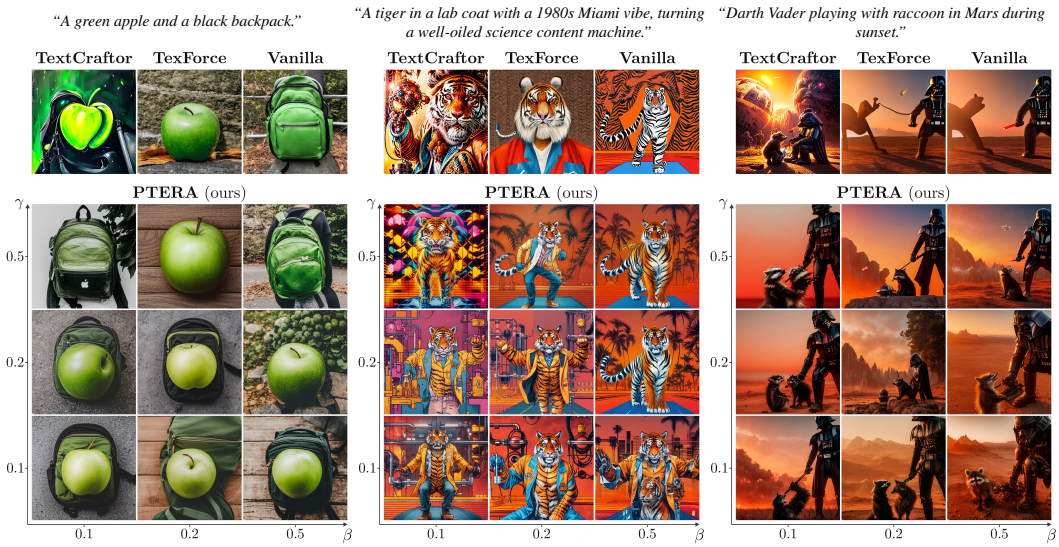
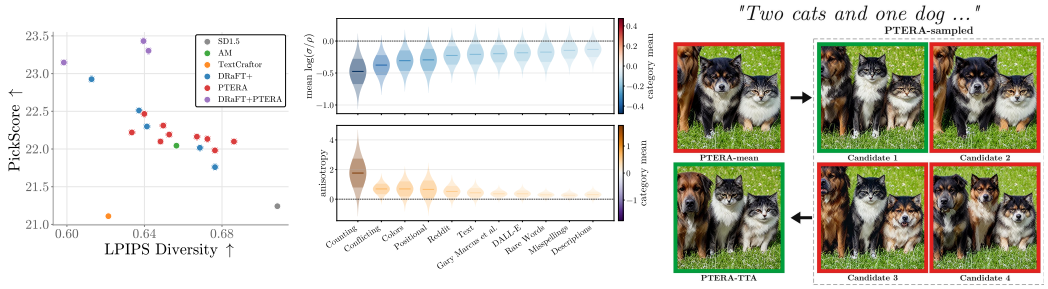


Figure 5: Qualitative comparison of PTERA and text encoder baselines across (β, γ). (Same random seed)



(a) Diversity-reward trade-off. (b) $\sigma_\varphi(c)$ analysis. (c) PTERA generations.

Figure 6: Analyses enabled by the probabilistic text encoder parameterization of PTERA. **(a)** Diversity-reward behavior of PTERA compared with reward-aligned baselines. **(b)** Learned scale structure of σ_φ across prompt categories. Top part shows magnitude, bottom part anisotropic shape of $\sigma_\varphi(c)$ against to reference ρ. **(c)** Sample variation induced by sampling from the learned PTERA text embedding distribution.

ure 4a shows that TextCraftor attains a high target reward but substantially degrades the remaining alignment metrics, revealing severe reward over-optimization (Specifically on AES). PTERA controls this trade-off through the diffusion and text embedding level regularization coefficients, allowing the target reward to improve while better preserving the behavior of the reference model.

Figure 4b illustrates how the learned text embedding distribution affects different inference variants. **PTERA-mean** conditions the diffusion model only on the learned mean embedding $\mu_\varphi(c)$, whereas **PTERA-sampled** draws a stochastic embedding from $p_\varphi(z \mid c)$ at the initial noise state. Thus, even with the same initial noise and deterministic sampler, PTERA-sampled introduces additional output diversity solely through text embedding sampling. **PTERA-TTA** then exploits this diversity through reward-guided selection, shifting the resulting samples toward higher reward. Additional results across broader settings are reported in Appendix D.1.

Parameter Sensitivity Analysis Figure 5 visualizes the effect of the two PTERA regularization coefficients, β and γ. Increasing either coefficient more strongly constrains the learned distribution toward the reference model. The grid in Figure 5 shows the smooth and interpretable trade-off between reward maximization and fidelity (We quantitatively examine the β-γ trade-off in Appendix D.2.). Figure 4c examines the sensitivity of PTERA to the reference exploration scale ρ. Across the values of ρ, neither PTERA-mean nor PTERA-sampled uniformly dominates the other. PTERA-TTA, however, consistently benefits from the learned distribution because its selection procedure can identify advantageous conditioning vectors for each denoising trajectory. Figure 4d further reports both the steering and held-out rewards as functions of the cumulative number of reward queries for a PTERA model trained on ImageReward.

Table 2: Quantitative results of PTERA on SDXL and SD3, PickScore steering on DrawBench.

Dataset		DrawBench					PartiPrompts				
Backbone	**Method**	**PS**	**IR**	**HPSv2**	**CLIP**	**AES**	**PS**	**IR**	**HPSv2**	**CLIP**	**AES**
SDXL	Base	22.219	0.329	0.251	**27.965**	5.896	22.231	0.576	0.265	27.574	6.229
	PTERA-mean	22.999	**0.773**	0.284	26.799	6.215	**23.207**	0.996	0.297	27.782	**6.456**
	PTERA-sampled	22.997	0.750	**0.285**	26.770	**6.224**	23.162	0.967	0.296	**27.822**	6.444
	PTERA-TTA	**23.000**	0.707	0.283	26.687	6.104	23.141	**1.000**	**0.298**	27.799	6.424
SD3	Base	22.494	0.926	0.283	**28.676**	5.955	22.576	1.103	0.297	27.967	6.234
	PTERA-mean	22.496	0.938	0.283	28.527	5.908	22.580	1.155	0.298	**28.081**	**6.249**
	PTERA-sampled	22.484	0.906	0.282	28.334	5.861	22.542	1.121	0.297	27.984	6.241
	PTERA-TTA	**22.532**	**0.987**	**0.284**	28.586	**5.965**	**22.605**	**1.197**	**0.300**	28.080	6.207

Table 3: Quantitative results demonstrating the complementary effect of PTERA when combined with DRaFT+. For GenEval, we generate four images per prompt and report the GenEval score for each compositional objective. For a fair comparison, we choose the DRaFT+ KL coefficient whose output diversity, measured by LPIPS, most closely matches that of PTERA. See Appendix C for details.

Dataset		DrawBench					GenEval						
Method	**Reward Queries**	**PS**	**IR**	**HPSv2**	**CLIP**	**AES**	**Single ↑**	**Two ↑**	**Count ↑**	**Color ↑**	**Position ↑**	**Color Attr. ↑**	**Overall ↑**
DRaFT+	20K	22.299	0.173	0.268	27.054	5.956	**0.988**	0.404	**0.403**	**0.787**	0.040	0.083	0.423
(+ DRaFT+)	(+20K)	22.593	0.221	0.274	26.707	6.060	0.984	0.409	0.372	0.766	0.040	0.083	0.422
(+ PTERA)	(+20K)	**23.432**	**0.530**	**0.287**	**27.082**	**6.137**	0.978	**0.508**	0.378	0.774	**0.075**	**0.113**	**0.453**

Exploiting PTERA distribution. Figure 6b shows that the learned scale $\sigma_\varphi(c)$ deviates from the reference ρ for different prompt categories, becoming more contracted and anisotropic for prompts with explicit constraints while remaining closer to the reference for open-ended descriptions. Figure 6c further shows that sampling from the learned distribution can recover correctly aligned object compositions missed by PTERA-mean, allowing PTERA-TTA to select a better-aligned image. See Appendix D.3 for the comparison to other rollout TTA methods in various experiment settings.

Scalability of PTERA. Beyond SD1.5, we evaluate PTERA on SDXL (Podell et al., 2024) and SD3 (Esser et al., 2024), which use different text encoder and diffusion architectures (See Table 5). Table 2 shows that PTERA remains effective across these backbones, indicating that the method is not tied to a particular architecture. Further qualitative results are provided in Appendix D.5.

We additionally investigate whether PTERA is complementary to diffusion parameter fine-tuning by combining it with DRaFT+. We compare allocating an additional reward query budget either to continued DRaFT+ optimization or to PTERA training on top of the resulting diffusion LoRA. Table 3 reports the corresponding GenEval performance. Beyond diffusion parameter fine-tuning, PTERA-TTA can also be integrated as a plug-in module into existing text-guided image editing methods (Ju et al., 2024; Huberman-Spiegelglas et al., 2024) or zero-shot transferability. See Appendix D.4 and D.2 that reports the corresponding quantitative and qualitative results.

Table 4: Computational cost comparison between DRaFT+ and PTERA under the same reward queries and the same batch size. All methods are evaluated under the same 4 L40S GPUs.

Method	**Architecture**	**Trainable Params. (M)**	**Peak VRAM (GiB)**	**Training Time (min)**	**FLOPs / Step (TFLOPs)**
DRaFT+	LoRA rank 3	1.122	35.41	480.62	91.44
PTERA	MHA Residual Block	**0.495**	**15.17**	**185.26**	**91.15**

Computational Cost. Table 4 compares PTERA with the DRaFT+ configuration used in Table 3 in terms of computational resources required during training. Because PTERA keeps the diffusion backbone frozen and updates only a compact text encoder residual module, it requires substantially fewer computational resources than LoRA training. This comparison demonstrates that PTERA provides a computational efficiency and a compatible extension to diffusion weight optimization.

5 Conclusion

In this work, we introduce PTERA, a probabilistic text conditioning framework for reward alignment in text-to-image diffusion models. PTERA learns a prompt-conditioned, reward-aware distribution around the deterministic embeddings produced by a pre-trained text encoder, while keeping the diffusion backbone frozen. The learned distribution improves reward-aligned generation and enables trajectory dependent exploration at test time through PTERA-TTA, yielding further gains at inference. PTERA also complements diffusion parameter fine-tuning and readily extends to other applications, from zero-shot transfer across backbones to text-guided image editing.

AI USE STATEMENT

We used generative AI tools to provide feedback on theoretical models and conceptual frameworks initially developed by the authors, to assist in refining author-proposed hypotheses, to assist with translation, and to clean and reformat datasets. Generative AI was also used as an auxiliary tool to organize and visualize experimental results, but not to interpret them.

The mathematical claims and all critical ingredients used to prove them were developed by the authors. The authors also wrote and verified all proofs without generative AI assistance. Research methodology and experiments were designed by the authors without generative AI assistance. Hypotheses were initially proposed by the authors, although generative AI was subsequently used to provide feedback and assist with their refinement. Generative AI was not used to conduct qualitative or thematic data analysis or to interpret the results. Synthetic-data generation was not applicable to this work.

All AI-assisted translations, data cleaning and reformatting, feedback on theoretical or conceptual frameworks, hypothesis refinement, and organization and visualization of results were reviewed and verified by the authors. All mathematical claims, proofs, methodological choices, experimental designs, and interpretations were also reviewed and verified by the authors.

We take responsibility for the final content of this work, including all text, claims, experiments, data processing, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY

A supplementary ZIP archive is provided with the submission, containing the code and configurations required to reproduce PTERA training and PTERA-TTA inference on SD1.5. During the discussion period, we will further release an anonymized repository containing source code for training PTERA and DTERA on SD1.5, SDXL, and SD3, the complete PTERA-TTA inference pipeline, all evaluation scripts, prompt lists, random seeds, and trained PTERA checkpoints. The repository will be made public upon publication.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629. IEEE, 2023.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Roni Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *International Conference on Learning Representations*, volume 2024, pp. 51304–51323, 2024.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, volume 2024, pp. 4965–4987, 2024.

Chaofeng Chen, Annan Wang, Haoning Wu, Liang Liao, Wenxiu Sun, Qiong Yan, and Weisi Lin. Enhancing diffusion models with text-encoder reinforcement learning. In *European Conference on Computer Vision*, pp. 182–198. Springer, 2024.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Sanghyuk Chun. Improved probabilistic image-text representations. In *International Conference on Learning Representations*, volume 2024, pp. 39787–39815, 2024.

Sanghyuk Chun, Wonjae Kim, Song Park, and Sangdoo Yun. Probabilistic language-image pre-training. In *International Conference on Learning Representations*, volume 2025, pp. 57190–57221, 2025.

Kevin Clark, Paul Vicol, Kevin Swersky, and David Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *International Conference on Learning Representations*, volume 2024, pp. 4793–4822, 2024.

Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. Multi-head attention: Collaborate instead of concatenate. *arXiv preprint arXiv:2006.16362*, 2020.

Carles Domingo i Enrich, Michal Drozdzal, Brian Karrer, and Ricky T. Q. Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 53791–53846, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/852f50969a9e523ec41d26f2f68bd456-Paper-Conference.pdf.

Monroe D Donsker and SR Srinivasa Varadhan. Asymptotic evaluation of certain markov process expectations for large time, i. *Communications on pure and applied mathematics*, 28(1):1–47, 1975.

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. In *Forty-first international conference on machine learning*, 2024.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in neural information processing systems*, 36:79858–79885, 2023.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=NAQvF08TcyG.

Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2426–2436, 2023.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems*, volume 36, pp. 52132–52152, 2023.

Yingqing Guo, Hui Yuan, Yukang Yang, Minshuo Chen, and Mengdi Wang. Gradient guidance for diffusion models: An optimization perspective. *Advances in neural information processing systems*, 37:90736–90770, 2024.

Yaru Hao, Zewen Chi, Li Dong, and Furu Wei. Optimizing prompts for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:66923–66939, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Peter Holderrieth, Uriel Singer, Tommi Jaakkola, Ricky TQ Chen, Yaron Lipman, and Brian Karrer. Glass flows: Efficient inference for reward alignment of flow and diffusion models. In *The Fourteenth International Conference on Learning Representations*, 2026.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Sujie Hu, Chubin Chen, Jiashu Zhu, Jiahong Wu, Xiangxiang Chu, and Xiu Li. Embedding-perturbed exploration preference optimization for flow models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=Ai4vdy80p5.

Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. An edit friendly ddpm noise space: Inversion and manipulations. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12469–12478. IEEE, 2024.

Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Pnp inversion: Boosting diffusion-based editing with 3 lines of code. In *International Conference on Learning Representations*, volume 2024, pp. 23395–23422, 2024.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022.

Mingyu Kim, Dongjun Kim, Amman Yusuf, Stefano Ermon, and Mijung Park. Training-free safe denoisers for safe use of diffusion models. *Advances in Neural Information Processing Systems*, 38:136004–136035, 2025.

Taehoon Kim, Henry Gouk, and Timothy Hospedales. Test-time alignment of text-to-image diffusion models via null-text embedding optimisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 43611–43620, 2026a.

Yeongmin Kim, Donghyeok Shin, Byeonghu Na, Minsang Park, Richard Lee Kim, and Il chul Moon. Lookahead sample reward guidance for test-time scaling of diffusion models. In *Forty-third International Conference on Machine Learning*, 2026b. URL https://openreview.net/forum?id=IbRm6gwmew.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 36652–36663, 2023.

Jaa-Yeon Lee, Yeobin Hong, Taesung Kwon, and Jong Chul Ye. Alignment-guided score matching for text-to-image alignment in diffusion models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=vKWxArobP3.

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gökcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, and Masatoshi Uehara. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=6QbbaEGkO7.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7985–7995. IEEE, 2024.

Zhen Liu, Tim Xiao, Carles Domingo i Enrich, Weiyang Liu, and Dinghuai Zhang. Value gradient guidance for flow matching alignment. *Advances in Neural Information Processing Systems*, 38: 11372–11404, 2025.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787, 2022.

Mengyao Lyu, Yuhong Yang, Haiwen Hong, Hui Chen, Xuan Jin, Yuan He, Hui Xue, Jungong Han, and Guiguang Ding. One-dimensional adapter to rule them all: Concepts, diffusion models and erasing applications. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7559–7568. IEEE, 2024.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6038–6047. IEEE, 2023.

Byeonghu Na, Minsang Park, Gyuwon Sim, Donghyeok Shin, HeeSun Bae, Mina Kang, Se Jung Kwon, Wanmo Kang, and Il chul Moon. Diffusion adaptive text embedding for text-to-image diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=cHi8QxGrZH.

XuanLong Nguyen. Convergence of latent mixing measures in finite and infinite mixture models. *The Annals of Statistics*, 41(1):370–400, 2013.

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.

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. In *International Conference on Learning Representations*, volume 2024, pp. 1862–1874, 2024.

Peter Potaptchik, Cheuk-Kit Lee, and Michael S Albergo. Tilt matching for scalable sampling and fine-tuning. *arXiv preprint arXiv:2512.21829*, 2025.

Peter Potaptchik, Adhi Saravanan, Abbas Mammadov, Alvaro Prat, Michael Samuel Albergo, and Yee Whye Teh. Meta flow maps enable scalable reward alignment. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=K5gV8Yptne.

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, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22500–22510. IEEE, 2023.

Seyedmorteza Sadat, Jakob Buhmann, Derek Bradley, Otmar Hilliges, and Romann Weber. Cads: Unleashing the diversity of diffusion models through condition-annealed sampling. In *International Conference on Learning Representations*, volume 2024, pp. 23723–23755, 2024.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, volume 35, pp. 36479–36494, 2022.

Christoph Schuhmann. CLIP+MLP Aesthetic Score Predictor, 2022. GitHub repository: christophschuhmann/improved-aesthetic-predictor.

Yichun Shi and Anil K Jain. Probabilistic face embeddings. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6902–6911, 2019.

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models. *arXiv preprint arXiv:2501.06848*, 2025.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

Narek Tumanyan, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Splicing vit features for semantic appearance transfer. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10738–10747. IEEE, 2022.

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1921–1930. IEEE, 2023.

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

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8228–8238. IEEE, 2024.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

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 *Advances in Neural Information Processing Systems*, volume 36, pp. 15903–15935, 2023.

Jianping Ye, Michel Wedel, and Kunpeng Zhang. Ipgo: Indirect prompt gradient optimization for parameter-efficient prompt-level fine-tuning on text-to-image models. *arXiv preprint arXiv:2503.21812*, 2025.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation. *Transactions on Machine Learning Research*, 2022.

Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 5907–5915, 2017.

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

Brian Nlong Zhao, Yuhang Xiao, Jiashu Xu, Xinyang Jiang, Yifan Yang, Dongsheng Li, Laurent Itti, Vibhav Vineet, and Yunhao Ge. Dreamdistribution: Learning prompt distribution for diverse in-distribution generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

A Theoretical Analyses

A.1 Detailed Process of Rollout TTA with text embedding exploration

For a candidate clean sample prediction $\hat{x}_0$, define its implied noise estimate and the deterministic DDIM update by

$$\hat{\epsilon}_t(x_t, \hat{x}_0) := \frac{x_t - \sqrt{\bar{\alpha}_t}\hat{x}_0}{\sqrt{1-\bar{\alpha}_t}}, \tag{14}$$

$$\mathrm{DDIMStep}(x_t, \hat{x}_0) := \sqrt{\bar{\alpha}_{t-1}}\hat{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}}\,\hat{\epsilon}_t(x_t, \hat{x}_0), \tag{15}$$

$$\mathrm{DDIM}_\theta(x_t, z) := \mathrm{DDIMStep}\left(x_t, \hat{x}^\theta_{0|t}(x_t, z)\right), \tag{16}$$

where $\bar{\alpha}_t$ denotes the cumulative diffusion coefficient. At a selection timestep $t \in \mathcal{S}$, Algorithm 1 samples candidate embeddings $z^{(i)}$, computes $\tilde{x}^{(i)}_{0|t} := \hat{x}^\theta_{0|t}(x_t, z^{(i)})$, and selects the candidate with the largest terminal reward. The selected prediction is then propagated through $\mathrm{DDIMStep}$. At $t \notin \mathcal{S}$, the standard update $\mathrm{DDIM}_\theta(x_t, z)$ is used, where $z = E_{\varphi_{\mathrm{ref}}}(c)$ for isotropic text embedding exploration and $z = \mu_\varphi(c)$ for PTERA-TTA. Thus, the greedy selection in Algorithm 1 can be viewed as the zero temperature (*i.e.* $\alpha \to 0$ in Equation 3), best-of-N variant of the reward weighted estimator in Equation 5.

Algorithm 1 Greedy selection toward reward tilted distribution with text embedding exploration

Require: text prompt c, text encoder E_φ, pre-trained diffusion model θ, reward function $\mathcal{R}$, number of candidates N, selection schedule $\mathcal{S} \subseteq \{T, \dots, 1\}$, exploration scaling factor $\rho > 0$

$x_T \sim \mathcal{N}(0, \mathbf{I})$ ▷ prior noise
for $t = T, \dots, 1$ **do**
  if $t \in \mathcal{S}$ **then** ▷ greedy selection step
    Generate N candidates $\{\hat{x}^{(i)}_{0|t}\}_{i=0}^{N-1}$ by:
    $\tilde{x}^i_{0|t} = \tilde{x}_{0|t}\big(x_t,\, z^i\big)$,
      where $z^i = E_\varphi(c) + \rho\,\varepsilon^i,\ \varepsilon^i \sim \mathcal{N}(0, \mathbf{I})$
      (or $z^i = \mu_\varphi(c) + \sigma_\varphi(c) \odot \varepsilon^i$, **PTERA-TTA**)
    $k \leftarrow \arg\max_{i \in \{0,\dots,N-1\}} \mathcal{R}\big(\tilde{x}^i_{0|t},\, c\big)$ ▷ greedy (best-of-N) selection
    $x_{t-1} \leftarrow \mathrm{DDIMStep}\big(x_t,\, \tilde{x}^k_{0|t}\big)$ ▷ Equation 15
  else
    $x_{t-1} \leftarrow \mathrm{DDIM}_\theta\big(x_t,\, E_\varphi(c)\big)$ ▷ standard denoising step
    (or $x_{t-1} \leftarrow \mathrm{DDIM}_\theta\big(x_t,\, \mu_\varphi(c)\big)$ on a fine-tuned environment.)
  end if
end for
return x_0

A.2 Theoretical Analysis of text embedding exploration

Proposition 1 ($\mathcal{W}_1$-stability under embedding perturbation)**.** Fix a text prompt c and a perturbation direction ε, and for $\rho \in \mathbb{R}_+$, let $E_c := E_\varphi(c)$ and $E^\rho_c := E_\varphi(c) + \rho\,\varepsilon$. Suppose the diffusion drift b_θ satisfies the following regularity conditions: for all $x, y, t \in [0, 1]$, and E,

(1) (Spatial Lipschitz continuity) $\big\|b_\theta(x, t, E) - b_\theta(y, t, E)\big\| \le L_x\|x - y\|$;

(2) (Smoothness in the embedding space) $\big\|\nabla_E\, b_\theta(x, t, E)\big\| \le L_c$.

Then the model distributions conditioned on E_c and E^ρ_c satisfy

$$\mathcal{W}_1\Big(p_\theta\big(x_0 \mid E_c\big),\ p_\theta\big(x_0 \mid E^\rho_c\big)\Big) \ \le\ \frac{L_c}{L_x}\big(e^{L_x} - 1\big)\,\rho\,\|\varepsilon\|. \tag{17}$$

Notation and scope of the proof. Throughout this proof we describe the sampler by its drift $b_\theta(x, t, E)$, a notation that appears only in this appendix. This formulation covers the stochastic reverse-diffusion sampler considered in the analysis; analogous bounds can be obtained for deterministic probability-flow dynamics under corresponding regularity conditions. For any concrete diffusion design, the sampling drift is an affine transformation with coefficients determined by the noise schedule of the learned network output. Consequently, the regularity conditions (1) and (2) of Proposition 1 imposed on b_θ translate into the corresponding Lipschitz and smoothness conditions on any of these parameterizations, up to schedule-dependent constants absorbed into L_x and L_c.

Proof. For brevity denote $E \coloneqq E_c$ and $E^\rho \coloneqq E_c^\rho = E_c + \rho\,\varepsilon$.

In the diffusion time convention, generation conditioned on an embedding E follows the reverse-time dynamics

$$dX_t^E = b_\theta\big(X_t^E, t, E\big)\,dt + \sigma_t\,d\bar{W}_t, \qquad X_1^E \sim \mathcal{N}(0, \mathbf{I}), \qquad t : 1 \to 0. \tag{18}$$

To avoid backward-time stochastic calculus, we change the time variable to $s \coloneqq 1 - t$ and set $Y_s^E \coloneqq X_{1-s}^E$, which solves the standard forward-time SDE

$$dY_s^E = \bar{b}_\theta\big(Y_s^E, s, E\big)\,ds + \bar{\sigma}_s\,dW_s, \qquad Y_0^E \coloneqq X_1^E \sim \mathcal{N}(0, \mathbf{I}), \qquad s \in [0, 1], \tag{19}$$

where $\bar{b}_\theta(y, s, E) \coloneqq -\,b_\theta(y, 1 - s, E)$, $\bar{\sigma}_s \coloneqq \sigma_{1-s}$, and W is a standard Brownian motion. Since negation and time relabeling do not affect the norms in conditions (1)-(2), the drift $\bar{b}_\theta$ satisfies the same regularity conditions with the same constants L_x, L_c; moreover, conditions (1)-(2) imply at most linear growth of the drift in y, so Equation 19 admits a pathwise unique strong solution on $[0, 1]$. The model distribution is the law of the terminal state, $p_\theta(x_0 \mid E) = \mathrm{Law}(Y_1^E)$.

On a common probability space, let $\xi \sim \mathcal{N}(0, \mathbf{I})$ be a single initial noise sample and W a single Brownian motion, independent of each other, and let the two processes share both:

$$dY_s^E = \bar{b}_\theta\big(Y_s^E, s, E\big)\,ds + \bar{\sigma}_s\,dW_s, \qquad Y_0^E = \xi, \tag{20}$$

$$dY_s^{E^\rho} = \bar{b}_\theta\big(Y_s^{E^\rho}, s, E^\rho\big)\,ds + \bar{\sigma}_s\,dW_s, \qquad Y_0^{E^\rho} = \xi. \tag{21}$$

Each process individually solves the sampling dynamics 19 with its own conditioning, so $Y_1^E \sim p_\theta(x_0 \mid E)$ and $Y_1^{E^\rho} \sim p_\theta(x_0 \mid E^\rho)$; hence the joint law of $\big(Y_1^E, Y_1^{E^\rho}\big)$ is a coupling of the two model distributions.

Define the pathwise error $e_s \coloneqq Y_s^{E^\rho} - Y_s^E$, with $e_0 = 0$. Writing Equation 20 and 21 in integral form and subtracting, the two stochastic integrals $\int_0^s \bar{\sigma}_r\,dW_r$ are identical and cancel exactly:

$$e_s = \int_0^s \Big[\bar{b}_\theta\big(Y_r^{E^\rho}, r, E^\rho\big) - \bar{b}_\theta\big(Y_r^E, r, E\big)\Big]\,dr. \tag{22}$$

The integrand is almost surely bounded on $[0, 1]$ (the coupled paths are continuous, hence bounded, and $\bar{b}_\theta$ is continuous), so $s \mapsto e_s$ is almost surely absolutely continuous with derivative $\dot{e}_s = \bar{b}_\theta(Y_s^{E^\rho}, s, E^\rho) - \bar{b}_\theta(Y_s^E, s, E)$ for a.e. s. This is the key consequence of the synchronous coupling: although each sample path is stochastic, the error between the coupled paths evolves by an ordinary differential equation and can be controlled deterministically.

Adding and subtracting $\bar{b}_\theta(Y_s^E, s, E^\rho)$ splits the error derivative into a *state difference* and a *text embedding difference*:

$$\dot{e}_s = \underbrace{\Big[\bar{b}_\theta\big(Y_s^{E^\rho}, s, E^\rho\big) - \bar{b}_\theta\big(Y_s^E, s, E^\rho\big)\Big]}_{\text{state difference}} + \underbrace{\Big[\bar{b}_\theta\big(Y_s^E, s, E^\rho\big) - \bar{b}_\theta\big(Y_s^E, s, E\big)\Big]}_{\text{text embedding difference}}. \tag{23}$$

The state difference is controlled by the spatial Lipschitz condition (1):

$$\Big\|\bar{b}_\theta\big(Y_s^{E^\rho}, s, E^\rho\big) - \bar{b}_\theta\big(Y_s^E, s, E^\rho\big)\Big\| \;\le\; L_x\,\Big\|Y_s^{E^\rho} - Y_s^E\Big\| = L_x\,\|e_s\|\,. \tag{24}$$

For the embedding difference, note that the mean value theorem does not hold verbatim for vector-valued maps; the correct statement is its integral form. By condition (2), the map $\tau \mapsto \bar{b}_\theta(y, s, E +$

$\tau\rho\varepsilon)$ is continuously differentiable, so the fundamental theorem of calculus along the segment from E to $E^\rho = E + \rho\varepsilon$ gives

$$\left\|\bar{b}_\theta\big(y, s, E^\rho\big) - \bar{b}_\theta\big(y, s, E\big)\right\| = \left\|\int_0^1 \nabla_E \bar{b}_\theta\big(y, s, E + \tau\rho\varepsilon\big)\big[\rho\,\varepsilon\big]\, d\tau\right\| \;\le\; L_c\,\rho\,\|\varepsilon\| \tag{25}$$

for every y and s, where the operator norm bound of condition (2) is applied under the integral sign. Combining Equation 23 to 25 with the triangle inequality,

$$\|\dot{e}_s\| \;\le\; L_x\,\|e_s\| + \rho\,L_c\,\|\varepsilon\| \qquad \text{for a.e. } s \in [0,1]. \tag{26}$$

Let $\phi(s) \coloneqq \|e_s\|$. Since $s \mapsto e_s$ is absolutely continuous and the norm is 1-Lipschitz, ϕ is absolutely continuous, and the reverse triangle inequality gives $|\phi'(s)| \le \|\dot{e}_s\|$ for a.e. s. Hence, by Equation 26, ϕ satisfies the differential inequality $\phi'(s) \le L_x\,\phi(s) + \rho L_c\,\|\varepsilon\|$ with $\phi(0) = 0$. Grönwall's Lemma now converts this differential inequality into an explicit bound. Multiplying both sides by the integrating factor $e^{-L_x s}$ shows that $\frac{d}{ds}\big(e^{-L_x s}\phi(s)\big) \le \rho L_c\,\|\varepsilon\|\; e^{-L_x s}$ almost everywhere, and integrating from 0 to s with $\phi(0) = 0$ and multiplying back by $e^{L_x s}$ yields

$$\|e_s\| \;\le\; \rho\,L_c\,\|\varepsilon\| \int_0^s e^{L_x(s-r)}\,dr \;=\; \rho\,\frac{L_c\,\|\varepsilon\|}{L_x}\big(e^{L_x s} - 1\big), \qquad s \in [0,1]. \tag{27}$$

The embedding difference injects an error at the constant rate $\rho L_c\,\|\varepsilon\|$, and each injected increment is amplified by at most $e^{L_x(s-r)}$ over the remaining time through the Lipschitz dynamics. Integrating the amplified increments gives Equation 27. Evaluating at $s = 1$, (*i.e.*, at diffusion time $t = 0$,)

$$\left\|Y_1^{E^\rho} - Y_1^{E}\right\| = \|e_1\| \;\le\; \rho\,\frac{L_c}{L_x}\big(e^{L_x} - 1\big)\,\|\varepsilon\| \quad \text{almost surely.} \tag{28}$$

The Wasserstein-1 distance is the infimum of the expected displacement over all couplings, $\mathcal{W}_1(\mu,\nu) = \inf_{\gamma\in\Pi(\mu,\nu)} \mathbb{E}_{(U,V)\sim\gamma}\,\|U - V\|$, therefore,

$$\mathcal{W}_1\Big(p_\theta\big(x_0 \mid E_c\big),\; p_\theta\big(x_0 \mid E_c^\rho\big)\Big) \;\le\; \mathbb{E}\left\|Y_1^{E^\rho} - Y_1^{E}\right\| \;\le\; \frac{L_c}{L_x}\big(e^{L_x} - 1\big)\,\rho\,\|\varepsilon\|\,, \tag{29}$$

is satisfied. □

A.3 Proof of Theorem 1

Proof of Theorem 1. Since the right-hand side of the claimed bound is an infimum over couplings, it suffices to show that, for an *arbitrary* coupling $q \in \mathcal{Q}(\Pi_c, \Pi_{c^{\mathrm{ref}}})$,

$$D_{\mathrm{KL}}\big(P_{\Pi_c} \,\big\|\, P_{\Pi_{c^{\mathrm{ref}}}}\big) \;\le\; \int D_{\mathrm{KL}}\big(p(x \mid z) \,\big\|\, p(x \mid z^{\mathrm{ref}})\big)\; q(dz, dz^{\mathrm{ref}}), \tag{30}$$

taking the infimum over q then yields the theorem.

Let $\mathcal{X}$ denote the space of denoising trajectories and let $\mathcal{G}$ be the class of bounded measurable functions $g : \mathcal{X} \to \mathbb{R}$. By the Donsker-Varadhan variational representation (Donsker & Varadhan, 1975) of the KL divergence, for any two probability measures P, Q on $\mathcal{X}$,

$$D_{\mathrm{KL}}(P \,\|\, Q) = \sup_{g\in\mathcal{G}} \Big\{\mathbb{E}_P[g] - \log \mathbb{E}_Q\big[e^g\big]\Big\}. \tag{31}$$

The utility of Equation 31 is that it expresses the KL divergence as a supremum of functionals that are *linear* in P and *convex* in Q, which is exactly the structure that survives mixing.

Both mixtures can be rewritten as integrals against the *same* index measure q:

$$P_{\Pi_c} = \int p(x \mid z)\, q(dz, dz^{\mathrm{ref}}), \quad P_{\Pi_{c^{\mathrm{ref}}}} = \int p(x \mid z^{\mathrm{ref}})\, q(dz, dz^{\mathrm{ref}}). \tag{32}$$

Indeed, integrating $p(x \mid z)$ against q only involves the first coordinate, whose marginal is Π_c.

Fix $g \in \mathcal{G}$. Using Equation 32 and Fubini's theorem,

$$\begin{aligned}
\mathbb{E}_{P_{\Pi_c}}[g] - \log \mathbb{E}_{P_{\Pi_{c^{\mathrm{ref}}}}}\left[e^g\right] &= \int \mathbb{E}_{p(x|z)}[g]\, q(dz, dz^{\mathrm{ref}}) - \log \int \mathbb{E}_{p(x|z^{\mathrm{ref}})}\left[e^g\right] q(dz, dz^{\mathrm{ref}}) \\
&\overset{(i)}{\leq} \int \mathbb{E}_{p(\cdot|z)}[g]\, q(dz, dz^{\mathrm{ref}}) - \int \log \mathbb{E}_{p(\cdot|z^{\mathrm{ref}})}\left[e^g\right] q(dz, dz^{\mathrm{ref}}) \\
&= \int \left(\mathbb{E}_{p(\cdot|z)}[g] - \log \mathbb{E}_{p(\cdot|z^{\mathrm{ref}})}\left[e^g\right]\right) q(dz, dz^{\mathrm{ref}}) \\
&\overset{(ii)}{\leq} \int \sup_{h \in \mathcal{G}} \left(\mathbb{E}_{p(\cdot|z)}[h] - \log \mathbb{E}_{p(\cdot|z^{\mathrm{ref}})}\left[e^h\right]\right) q(dz, dz^{\mathrm{ref}}) \\
&= \int D_{\mathrm{KL}}\big(p(\cdot \mid z) \,\big\|\, p(\cdot \mid z^{\mathrm{ref}})\big)\; q(dz, dz^{\mathrm{ref}}), \qquad (33)
\end{aligned}$$

where $p(\cdot \mid z)$ abbreviates the kernel $p(x \mid z)$. Inequality (i) is Jensen's inequality applied to the concave function $\log$. Inequality (ii) holds because the integrand, evaluated at the particular function g, is dominated by its supremum over $h \in \mathcal{G}$ pointwise in (z, z^{ref}). The final equality applies the Donsker-Varadhan representation once more, now to each pair of conditional distributions $\big(p(x \mid z),\, p(x \mid z^{\mathrm{ref}})\big)$.

The right-hand side of Equation 33 does not depend on g. Taking the supremum over $g \in \mathcal{G}$ on the left-hand side and invoking Equation 31 for the pair $(P_{\Pi_c}, P_{\Pi_{c^{\mathrm{ref}}}})$ yields Equation 30. Finally, taking the infimum over all couplings gives

$$D_{\mathrm{KL}}\big(P_{\Pi_c} \,\big\|\, P_{\Pi_{c^{\mathrm{ref}}}}\big) \;\leq\; \inf_{q \in \mathcal{Q}(\Pi_c, \Pi_{c^{\mathrm{ref}}})} \int D_{\mathrm{KL}}\big(p(x \mid z) \,\big\|\, p(x \mid z^{\mathrm{ref}})\big)\; q(dz, dz^{\mathrm{ref}}). \qquad (34)$$

When applying the theorem to diffusion trajectories (Fan et al., 2023), we can instantiate a sample x as the corresponding diffusion trajectory $x_{0:T}$. □

A.4 Derivation of the Surrogate Regularizations $\mathcal{D}_{\mathrm{diff}}$ and $\mathcal{D}_{\mathrm{text}}$

Throughout this section, $p_\theta(x_{0:T} \mid z)$ denotes the (frozen) diffusion kernel, and let $\Pi_c = p_\varphi(\cdot \mid c)$ and $\Pi_{c^{\mathrm{ref}}} = p_{\varphi_{\mathrm{ref}}}(\cdot \mid c)$ be the target and reference mixing measures of Theorem 1, with mixtures $P_{\Pi_c} = \int p_\theta(x_{0:T} \mid z)\, \Pi_c(dz)$ and $P_{\Pi_{c^{\mathrm{ref}}}} = \int p_\theta(x_{0:T} \mid z^{\mathrm{ref}})\, \Pi_{c^{\mathrm{ref}}}(dz^{\mathrm{ref}})$. We derive the two upper bounds

$$D_{\mathrm{KL}}\big(P_{\Pi_c} \,\big\|\, P_{\Pi_{c^{\mathrm{ref}}}}\big) \;\leq\; \mathcal{D}_{\mathrm{diff}}(\varphi; c) \quad \text{and} \quad D_{\mathrm{KL}}\big(P_{\Pi_c} \,\big\|\, P_{\Pi_{c^{\mathrm{ref}}}}\big) \;\leq\; \mathcal{D}_{\mathrm{text}}(\varphi; c), \qquad (35)$$

which justify using either term. Hence any linear combination $\beta\mathcal{D}_{\mathrm{diff}} + \gamma\mathcal{D}_{\mathrm{text}}$ with $\beta, \gamma \geq 0$ operates as a surrogate regularizer for the reverse KL divergence in Definition 1.

Derivation of $\mathcal{D}_{\mathrm{diff}}$ (shared-ε matching). Let $\varepsilon \sim \mathcal{N}(0, \mathbf{I})$ and consider the pair

$$(z, z^{\mathrm{ref}}) \;=\; \big(z_\varphi(c, \varepsilon),\; z_{\varphi_{\mathrm{ref}}}(c, \varepsilon)\big), \qquad (36)$$

i.e., both encoders are evaluated on the *same* noise instantiation. By the reparameterization in Equation 9, each coordinate of the pair is marginally distributed as the corresponding encoder output. $z \sim p_\varphi(\cdot \mid c) = \Pi_c$ and $z^{\mathrm{ref}} \sim p_{\varphi_{\mathrm{ref}}}(\cdot \mid c) = \Pi_{c^{\mathrm{ref}}}$. Hence the joint distribution of the pair 36, denoted $\bar{q}$, is an element of $\mathcal{Q}(\Pi_c, \Pi_{c^{\mathrm{ref}}})$. Since the right-hand side of Theorem 1 is an infimum over all couplings, instantiating it at $\bar{q}$ yields

$$\begin{aligned}
D_{\mathrm{KL}}\big(P_{\Pi_c} \,\big\|\, P_{\Pi_{c^{\mathrm{ref}}}}\big) \;&\leq\; \int D_{\mathrm{KL}}\big(p_\theta(\cdot \mid z) \,\big\|\, p_\theta(\cdot \mid z^{\mathrm{ref}})\big)\; \bar{q}(dz, dz^{\mathrm{ref}}) \\
&=\; \mathbb{E}_{\varepsilon \sim \mathcal{N}(0, \mathbf{I})}\Big[D_{\mathrm{KL}}\big(p_\theta(\cdot \mid z) \,\big\|\, p_\theta(\cdot \mid z^{\mathrm{ref}})\big)\Big] \;=\; \mathcal{D}_{\mathrm{diff}}(\varphi; c), \qquad (37)
\end{aligned}$$

where, (z, z^{ref}) is the pair defined in Equation 36 evaluated at shared-ε. Integrating a function of (z, z^{ref}) against $\bar{q}$ is the same as taking the expectation over ε of the function evaluated at $\big(z_\varphi(c, \varepsilon), z_{\varphi_{\mathrm{ref}}}(c, \varepsilon)\big)$.

Under the standard DDPM (Ho et al., 2020) parameterization, the conditional Gaussian KL terms in $\mathcal{D}_{\text{diff}}$ reduce to a closed-form weighted denoising discrepancy. Let $\alpha_t \coloneqq 1-\beta_t$ and $\bar{\alpha}_t \coloneqq \prod_{s=1}^{t} \alpha_s$, and write the reverse transition as

$$p_\theta(x_{t-1} \mid x_t, z) = \mathcal{N}\left(x_{t-1}; \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\hat{x}^\theta_{0|t}(x_t, z) + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t, \sigma_t^2\mathbf{I}\right), \tag{38}$$

where $\sigma_t^2 > 0$ is fixed by the DDPM noise schedule and is independent of x_t and z. Since the two reverse kernels conditioned on z and z^{ref} share the same covariance,

$$D_{\text{KL}}\big(p_\theta(x_{t-1} \mid x_t, z) \,\big\|\, p_\theta(x_{t-1} \mid x_t, z^{\text{ref}})\big) = \frac{\bar{\alpha}_{t-1}\beta_t^2}{2\sigma_t^2(1-\bar{\alpha}_t)^2}\left\|\hat{x}^\theta_{0|t}(x_t, z) - \hat{x}^\theta_{0|t}(x_t, z^{\text{ref}})\right\|_2^2. \tag{39}$$

Therefore, applying the chain rule for KL divergence to the DDPM trajectories yields

$$\mathcal{D}_{\text{diff}}(\varphi; c) = \mathbb{E}_{\varepsilon\sim\mathcal{N}(0,\mathbf{I})}\mathbb{E}_{x_{0:T}\sim p_\theta(\cdot|z)}\left[\sum_{t=1}^{T}\frac{\bar{\alpha}_{t-1}\beta_t^2}{2\sigma_t^2(1-\bar{\alpha}_t)^2}\left\|\hat{x}^\theta_{0|t}(x_t, z) - \hat{x}^\theta_{0|t}(x_t, z^{\text{ref}})\right\|_2^2\right], \tag{40}$$

where $(z, z^{\text{ref}}) = \big(z_\varphi(c, \varepsilon), z_{\varphi_{\text{ref}}}(c, \varepsilon)\big)$.

Derivation of $\mathcal{D}_{\text{text}}$ (shared-z matching). This bound couples the two models at the level of the *joint* distributions over $(z, x_{0:T})$ that share the same latent z and the same diffusion kernel,

$$J_\varphi \coloneqq p_\varphi(z \mid c)\, p_\theta(x_{0:T} \mid z) \quad \text{and} \quad J_{\varphi_{\text{ref}}} \coloneqq p_{\varphi_{\text{ref}}}(z \mid c)\, p_\theta(x_{0:T} \mid z). \tag{41}$$

Marginalizing a joint distribution over z is an application of one and the same Markov kernel to both arguments, so by the data processing inequality,

$$D_{\text{KL}}\big(P_{\Pi_c} \,\big\|\, P_{\Pi_{c^{\text{ref}}}}\big) \;\le\; D_{\text{KL}}(J_\varphi \,\|\, J_{\varphi_{\text{ref}}})\,. \tag{42}$$

The decomposition of the KL divergence, applied to the joints in the Markov order $z \to x_{0:T}$ as

$$\begin{aligned} D_{\text{KL}}(J_\varphi \,\|\, J_{\varphi_{\text{ref}}}) \;&=\; D_{\text{KL}}(p_\varphi(z \mid c) \,\|\, p_{\varphi_{\text{ref}}}(z \mid c)) + \mathbb{E}_{z\sim p_\varphi(\cdot|c)}\underbrace{D_{\text{KL}}(p_\theta(\cdot \mid z) \,\|\, p_\theta(\cdot \mid z))}_{=0} \\ &=\; \mathcal{D}_{\text{text}}(\varphi; c), \end{aligned} \tag{43}$$

where the conditional term vanishes identically because the two joints share the same diffusion kernel $p_\theta(x_{0:T} \mid z)$. Combining equation 42 and equation 43 gives the second bound in equation 35.

Finally, since the generated sample x_0 is a coordinate projection of the trajectory $x_{0:T}$, one further application of the data processing inequality gives $D_{\text{KL}}(p_{\theta,\varphi}(x_0 \mid c) \,\|\, p_{\theta,\varphi_{\text{ref}}}(x_0 \mid c)) \le D_{\text{KL}}\big(P_{\Pi_c} \,\big\|\, P_{\Pi_{c^{\text{ref}}}}\big)$ (Fan et al., 2023), so both regularizers also control the KL divergence between the distributions of the final samples.

Since both encoder distributions are diagonal Gaussians, $\mathcal{D}_{\text{text}}$ itself admits the following closed form. Let

$$\mu = \mu_\varphi(c), \qquad \sigma = \sigma_\varphi(c), \qquad \mu' = E_{\varphi_{\text{ref}}}(c). \tag{44}$$

Then

$$\mathcal{D}_{\text{text}}(\varphi; c) = \frac{1}{2}\sum_{j=1}^{d}\left[\log\frac{\rho^2}{\sigma_j^2} + \frac{\sigma_j^2 + (\mu_j - \mu'_j)^2}{\rho^2} - 1\right], \tag{45}$$

where ρ^2 is variance of reference text embedding distribution $p_{\theta_{\text{ref}},\varphi_{\text{ref}}}$.

A.5 Justification for Linear Combination of Two Different Regularizations

In this section, we discuss the necessity of each of the two regularizations $\mathcal{D}_{\text{diff}}$ and $\mathcal{D}_{\text{text}}$ by proposing a simulated environment (which is motivated by the assumptions in Guo et al. (2024)) consisting of a simple Gaussian generator θ and Gaussian text encoders $\varphi, \varphi_{\text{ref}}$. In this environment the true marginal KL divergence admits a closed form, so the below Proposition 2 tractably characterizes the ratio between the true KL divergence and each of $\mathcal{D}_{\text{diff}}$ and $\mathcal{D}_{\text{text}}$, where the two regularizations of Definition 1 are understood with the one-step generator $p_\theta(x \mid z)$ in place of the diffusion kernel $p_\theta(x_{0:T} \mid z)$.

Simulated environment. Given a text prompt c, the target and reference text encoders are modeled as

$$p_\varphi(z \mid c) = \mathcal{N}\big(\mu,\, \mathrm{diag}(\sigma^2)\big), \qquad p_{\varphi_{\mathrm{ref}}}(z \mid c) = \mathcal{N}\big(0,\, \rho^2 I\big), \tag{46}$$

and the generator model as

$$p_\theta(x \mid z) = \mathcal{N}\big(Az,\, \tau^2 I\big), \qquad A \in \mathbb{R}^{D\times d}, \quad x \in \mathbb{R}^D, \quad z \in \mathbb{R}^d, \tag{47}$$

with singular value decomposition $A = U\Lambda V^\top$ and nonnegative singular values $a_1, \ldots, a_{\min(D,d)}$. The marginal (mixture) distribution is

$$p_{\theta,\varphi}(x \mid c) \coloneqq \int p_\theta(x \mid z)\, p_\varphi(z \mid c)\, dz, \tag{48}$$

defined analogously for φ_{ref}, and the true marginal KL divergence is denoted by

$$\mathrm{KL} \;\coloneqq\; D_{\mathrm{KL}}(p_{\theta,\varphi}(x \mid c) \,\|\, p_{\theta,\varphi_{\mathrm{ref}}}(x \mid c))\,. \tag{49}$$

Proposition 2. In the simulated environment of Equation 46 to 49, suppose the target encoder deviates from the reference encoder by a mean shift along the i-th right singular direction of A at the reference scale, *i.e.*,

$$\mu = \delta\, v_i \;\; (\delta \neq 0), \qquad \sigma = \rho \mathbf{1}, \tag{50}$$

where v_i is the i-th column of V and $a_i > 0$. Then the true marginal KL divergence is

$$\mathrm{KL} = \frac{a_i^2 \delta^2}{2\,(a_i^2\rho^2 + \tau^2)}, \tag{51}$$

and the two regularizations satisfy

$$\frac{\mathcal{D}_{\mathrm{diff}}}{\mathrm{KL}} = 1 + \kappa_i, \qquad \frac{\mathcal{D}_{\mathrm{text}}}{\mathrm{KL}} = 1 + \frac{1}{\kappa_i}, \qquad \text{where } \; \kappa_i \coloneqq \frac{a_i^2\rho^2}{\tau^2}. \tag{52}$$

Proof. Since $A \in \mathbb{R}^{D\times d}$ is rectangular, its singular value decomposition reads $A = U\Lambda V^\top$ with orthogonal matrices $U \in \mathbb{R}^{D\times D}$, $V \in \mathbb{R}^{d\times d}$ and a rectangular diagonal matrix $\Lambda \in \mathbb{R}^{D\times d}$, $\Lambda_{jj} = a_j$ for $j \le \min(D,d)$ and zero elsewhere. Define the rotated variables

$$\tilde{x} \coloneqq U^\top x, \qquad \tilde{z} \coloneqq V^\top z, \qquad \tilde{\mu} \coloneqq V^\top \mu. \tag{53}$$

We notate how each model component transforms under Equation 53. *(i) Target encoder.* If $z \sim \mathcal{N}(\mu, \rho^2\mathbf{I})$, then $\tilde{z} \sim \mathcal{N}\big(\tilde{\mu},\, \rho^2 V^\top V\big) = \mathcal{N}\big(\tilde{\mu},\, \rho^2\mathbf{I}\big)$. *(ii) Reference encoder.* By the same computation with $\mu = 0$ and covariance $\rho^2\mathbf{I}$, the rotated reference is again $\mathcal{N}(0, \rho^2\mathbf{I})$. *(iii) Generator.* Conditioned on z, we have $x \sim \mathcal{N}(Az, \tau^2\mathbf{I})$, hence

$$\tilde{x} = U^\top x \sim \mathcal{N}\big(U^\top Az,\; \tau^2 U^\top U\big) = \mathcal{N}\big(\Lambda V^\top z,\; \tau^2\mathbf{I}\big) = \mathcal{N}\big(\Lambda\tilde{z},\; \tau^2\mathbf{I}\big), \tag{54}$$

using $U^\top A = \Lambda V^\top$ and the orthogonality of U. *(iv) Invariance of all divergences.* The KL divergence is invariant under applying one and the same bijection to both of its arguments; rotating $x \mapsto U^\top x$ leaves KL and each conditional KL inside $\mathcal{D}_{\mathrm{diff}}$ unchanged, and rotating $z \mapsto V^\top z$ leaves $\mathcal{D}_{\mathrm{text}}$ unchanged. We may therefore compute all three quantities in the rotated coordinates, where the model is fully coordinate-separable:

$$\tilde{x}_j = a_j\tilde{z}_j + \tilde{w}_j \;\; (j \le \min(D,d)), \qquad \tilde{x}_j = \tilde{w}_j \;\; (j > d \text{ if } D > d), \tag{55}$$

with $\tilde{w} \sim \mathcal{N}(0, \tau^2 I_D)$, while the coordinates $\tilde{z}_j$ with $j > D$ (if $d > D$) do not enter $\tilde{x}$ at all.

Under the mean shift $\mu = \delta v_i$ with $\sigma = \rho\mathbf{1}$, the rotated mean is $\tilde{\mu} = V^\top(\delta v_i) = \delta e_i$, where e_i is the i-th standard basis vector; thus in the rotated coordinates the target encoder is $\mathcal{N}(\delta e_i, \rho^2\mathbf{I})$ and the reference encoder is $\mathcal{N}(0, \rho^2\mathbf{I})$.

By Equation 55, the rotated marginals are Gaussians with diagonal covariances. Integrating $\tilde{z}$ out of Equation 54,

$$\tilde{x} \sim \mathcal{N}\big(\delta a_i e_i,\; C\big) \qquad \text{(target distribution)}, \tag{56}$$

$$\tilde{x} \sim \mathcal{N}\big(0,\; C\big) \qquad \text{(reference distribution)}, \tag{57}$$

where $C \coloneqq \Lambda(\rho^2\mathbf{I})\Lambda^\top + \tau^2\mathbf{I} = \operatorname{diag}\big(a_j^2\rho^2 + \tau^2\big)_{j=1}^{D}$ (with $a_j \coloneqq 0$ for $j > \min(D, d)$). The two marginals share the same covariance and differ only in the mean, so by the Gaussian KL formula for equal covariances,

$$D_{\mathrm{KL}}(\mathcal{N}(m_1, C) \,\|\, \mathcal{N}(m_2, C)) = \tfrac{1}{2}\,(m_1 - m_2)^\top C^{-1}(m_1 - m_2), \tag{58}$$

$$\mathrm{KL} = \frac{(\delta a_i)^2}{2\,(a_i^2\rho^2 + \tau^2)} = \frac{a_i^2\delta^2}{2\,(a_i^2\rho^2 + \tau^2)}, \tag{59}$$

which proves the claimed closed form of the true marginal KL.

Computation of $\mathcal{D}_{\mathrm{diff}}$. Under the shared-$\varepsilon$ matching, the coupled embeddings are $z = \mu + \rho\,\varepsilon$ and $z^{\mathrm{ref}} = \rho\,\varepsilon$, so their difference is deterministic: $z - z^{\mathrm{ref}} = \mu$. Both conditionals are Gaussians with the same covariance $\tau^2 I$, hence

$$\mathcal{D}_{\mathrm{diff}} = \mathbb{E}_\varepsilon\left[\frac{\|Az - Az^{\mathrm{ref}}\|^2}{2\tau^2}\right] = \frac{\|A\mu\|^2}{2\tau^2} = \frac{\delta^2\|\Lambda e_i\|^2}{2\tau^2} = \frac{a_i^2\delta^2}{2\tau^2}, \tag{60}$$

using $\|A\mu\| = \|U\Lambda V^\top(\delta v_i)\| = \delta\|\Lambda e_i\| = \delta a_i$ by the orthogonality of U and V. Dividing Equation 60 by Equation 59,

$$\frac{\mathcal{D}_{\mathrm{diff}}}{\mathrm{KL}} = \frac{a_i^2\rho^2 + \tau^2}{\tau^2} = 1 + \kappa_i. \tag{61}$$

Computation of $\mathcal{D}_{\mathrm{text}}$. The two encoders are Gaussians with the same covariance $\rho^2 I$ and mean difference μ, so by the same equal-covariance formula equation 58,

$$\mathcal{D}_{\mathrm{text}} = D_{\mathrm{KL}}\big(\mathcal{N}(\mu, \rho^2 I) \,\big\|\, \mathcal{N}(0, \rho^2 I)\big) = \frac{\|\mu\|^2}{2\rho^2} = \frac{\delta^2}{2\rho^2}, \tag{62}$$

and dividing by equation 59,

$$\frac{\mathcal{D}_{\mathrm{text}}}{\mathrm{KL}} = \frac{a_i^2\rho^2 + \tau^2}{a_i^2\rho^2} = 1 + \frac{1}{\kappa_i}, \tag{63}$$

which completes the proof. □

Proposition 2 clarifies the complementary tightness of the two surrogate regularizers. Since $\kappa_i > 0$, Equation 52 implies

$$\mathrm{KL} < \min\{\mathcal{D}_{\mathrm{diff}}, \mathcal{D}_{\mathrm{text}}\} = \mathrm{KL}\left(1 + \min\{\kappa_i, \kappa_i^{-1}\}\right) \le 2\,\mathrm{KL}, \tag{64}$$

where the upper equality is attained when $\kappa_i = 1$. Thus, $\mathcal{D}_{\mathrm{diff}}$ is tighter when the generator is relatively insensitive to the corresponding embedding direction ($\kappa_i < 1$), whereas $\mathcal{D}_{\mathrm{text}}$ is tighter when that direction is strongly propagated through the generator ($\kappa_i > 1$). The factor-two guarantee in Equation 64 applies to the better of the two bounds and does not generally extend to an arbitrary fixed convex combination. Nevertheless, because both terms individually upper bound the true marginal KL divergence, every convex combination $\mathcal{D}_\lambda = \lambda\mathcal{D}_{\mathrm{diff}} + (1-\lambda)\mathcal{D}_{\mathrm{text}}$, $\lambda \in [0, 1]$, remains a valid upper bound. Setting $\lambda = \beta/(\beta+\gamma)$ and $\alpha = \beta + \gamma$ gives $\beta\mathcal{D}_{\mathrm{diff}} + \gamma\mathcal{D}_{\mathrm{text}} = \alpha\mathcal{D}_\lambda$, which upper bound of $\alpha\,\mathrm{KL}$. We therefore use this weighted combination in practice, allowing the two regularizers to complement each other.

B PTERA AS A PROBABILISTIC EXTENSION OF DTERA

DTERA and PTERA use the same frozen diffusion model, but differ in the class of conditional image distributions that can be induced by the text encoder. DTERA assigns a single text embedding to each prompt, such that its trainable parameters select one conditional distribution of the frozen diffusion model. In contrast, PTERA predicts a distribution over text embeddings and marginalizes the corresponding conditional image distributions. The probabilistic parameterization should therefore be understood as an enlargement of the attainable output distribution family (Chun, 2024; Shi & Jain, 2019). The following proposition formalizes this representational relation:

Proposition 3. For a prompt c and a frozen diffusion model p_θ, let Φ_{D} and Φ_{P} denote the parameter sets of DTERA and PTERA, respectively. Define their attainable conditional image distribution families as

$$\mathcal{P}_{\mathrm{D}}(c) := \{p_\theta(x_0 \mid E_{\varphi_{\mathrm{D}}}(c)) : \varphi_{\mathrm{D}} \in \Phi_{\mathrm{D}}\}, \tag{65}$$

$$\mathcal{P}_{\mathrm{P}}(c) := \left\{ \int p_\theta(x_0 \mid z)\, \Pi_c^{\varphi_{\mathrm{P}}}(dz) : \varphi_{\mathrm{P}} \in \Phi_{\mathrm{P}} \right\}, \tag{66}$$

where $\Pi_c^{\varphi_{\mathrm{P}}} = \mathcal{N}\big(\mu_{\varphi_{\mathrm{P}}}(c), \mathrm{diag}\big(\sigma^2_{\varphi_{\mathrm{P}}}(c)\big)\big)$. Suppose that the mean branch of PTERA can represent every deterministic embedding attainable by DTERA and that the zero scale is admissible; namely, for every $\varphi_{\mathrm{D}} \in \Phi_{\mathrm{D}}$, there exists $\varphi_{\mathrm{P}} \in \Phi_{\mathrm{P}}$ such that $\mu_{\varphi_{\mathrm{P}}}(c) = E_{\varphi_{\mathrm{D}}}(c), \sigma_{\varphi_{\mathrm{P}}}(c) = 0$. Interpreting the resulting zero-variance Gaussian as a point mass distribution, we have $\mathcal{P}_{\mathrm{D}}(c) \subseteq \mathcal{P}_{\mathrm{P}}(c)$.

Proposition 3 shows that DTERA is recovered as the zero-variance special case of PTERA. The mean $\mu_\varphi(c)$ retains the deterministic conditioning solutions available to DTERA, while a nonzero $\sigma_\varphi(c)$ additionally induces mixtures of conditional image distributions that need not be representable by any single text embedding. Consequently, for an objective evaluated on the induced image distribution, the best attainable value over the full PTERA family cannot be lower than that over DTERA under exact optimization.

C Experimental Settings

Datasets. We use DrawBench (Saharia et al., 2022) as the prompt-only training set and as an in-domain evaluation benchmark. DrawBench contains 200 prompts organized into 11 categories that probe a broad range of text-to-image generation capabilities, including color rendering, object counting, spatial relations, text generation, rare words, misspellings, conflicting object interactions, and complex descriptions. Since all reward-based fine-tuning methods considered in this work require only textual prompts and differentiable or queryable reward functions, no paired training images are used.

For held-out evaluation, we use the HPSv2, PartiPrompts, and GenEval prompt sets. The HPSv2 evaluation benchmark (Wu et al., 2023) contains 3,200 prompts, with 800 prompts in each of four visual-style categories: *Animation*, *Concept Art*, *Painting*, and *Photo*. The first three subsets are constructed from de-biased and rewritten community prompts, whereas the Photo subset is based on COCO Captions. This benchmark is designed to provide a statistically stable and less style-biased evaluation of human preference across heterogeneous image distributions. PartiPrompts (Yu et al., 2022) consists of 1,632 English prompts released with the Parti model. The prompts span 12 semantic categories and 11 challenge types, ranging from basic descriptions to complex compositions involving world knowledge, fine-grained attributes, quantities, spatial relationships, writing, and artistic styles. Finally, GenEval (Ghosh et al., 2023) contains 553 templated prompts divided into six object-centric tasks: *single object*, *two objects*, *counting*, *colors*, *position*, and *attribute binding*. Following the standard GenEval protocol, we generate four images per prompt and average the resulting binary correctness scores over all images and prompts.

Baseline Configurations. Unless otherwise specified, the reproduced methods use the common sampling and optimization settings described in the main text, including a classifier-free guidance scale of 7.5 and 100 training epochs. For TextCraftor (Li et al., 2024), we retain the original prompt-based training configuration: images are generated using a 25-step DDIM rollout, gradients are backpropagated through the final five denoising transitions, and the text encoder is optimized with a learning rate of 10^{-6}. At evaluation time, however, we use the common 50-step sampler and condition every denoising transition on the output of the fine-tuned text encoder. This differs from the original TextCraftor protocol, which applies the fine-tuned encoder to only 15 of the 25 evaluation steps, and provides an all-step conditioning protocol consistent with that used for PTERA.

Adjoint Matching (Domingo i Enrich et al., 2025) (AM) parameterizes the trade-off between reward maximization and reference preservation by multiplying the reward by a scalar λ, rather than placing a coefficient directly in front of the regularization term. Among the configurations considered in its official implementation, we reproduce the $\lambda = 100$ setting, which yields the highest steering reward

in our experiments.[1] For TexForce (Chen et al., 2024), we apply rank-16 LoRA adaptation to the text encoder, resulting in approximately 1.17M trainable parameters, and use a learning rate of 3×10^{-4}.

We implement DRaFT+ as the reference-regularized extension of DRaFT (Clark et al., 2024) provided by NVIDIA NeMo Aligner.[2] We use a learning rate of 10^{-4} and reduce the diffusion-model LoRA rank from the default value of 32 to 3, bringing its trainable parameter count into the same range as that of PTERA. For the reward-diversity analysis in Figure 6a, we evaluate five reference-regularization coefficients, $\{0.2, 0.5, 1, 2, 4\}$, while holding the remaining training configuration fixed. Table 3 reports the results for a DRaFT+ KL coefficient of 1.0, which most closely matches PTERA in output diversity. Under an equal budget of 20k reward queries, PTERA and DRaFT+ achieve LPIPS diversities of 0.640 and 0.641, respectively.

PTERA training configurations. Our implementation of PTERA follows the reward-backpropagation training pipeline used by the DRaFT+ implementation in NVIDIA NeMo Aligner. Motivated by the empirical effectiveness of truncated reward backpropagation in DRaFT (Clark et al., 2024), we generate each training sample using a 50-step DDIM trajectory but backpropagate the reward gradient through only the final denoising transition. We use this one-step truncation for both PTERA and DTERA throughout the main experiments. Detailed architecture and optimization configurations for each model backbone are provided in Table 5.

TTA configurations. Unless otherwise specified, the PTERA-TTA results reported in Figure 2b, Table 1, Figure 6c, and Table 2 use the selection schedule $\mathcal{S} = \{50, 40, 30, 20, 10\}$ in Algorithm 1, defined over the 50-step DDIM sampling trajectory, with $N = 5$ candidates (except Figure 2b) at each selection step. The additional comparisons with alternative rollout TTA methods in Appendix D.3 use the method-specific configurations reported in Table 9.

LPIPS Metrics. We use LPIPS in two distinct roles: measuring deviation from a corresponding vanilla sample and measuring diversity among samples generated for the same prompt. In Figure 2b, each steered image is paired with the vanilla image generated from the original text embedding under the same prompt and initial noise. The reported LPIPS distance quantifies the perceptual change induced by reward steering, whereas the win rate is the fraction of paired examples for which the steered image receives a higher PickScore than its vanilla counterpart. Thus, lower LPIPS indicates stronger preservation of the original sample in this experiment.

For diversity evaluation, let $\{x_c^{(i)}\}_{i=1}^K$ denote K images generated from a prompt c. We compute the prompt-wise mean pairwise LPIPS as

$$\frac{2}{K(K-1)} \sum_{1 \le i < j \le K} \mathrm{LPIPS}\left(x_c^{(i)}, x_c^{(j)}\right) \tag{67}$$

For Figure 4b, we generate $K = 16$ images for every DrawBench prompt, compute Equation 67 within each prompt, and then average the resulting values over all 200 prompts. For Figure 6a, we randomly select one DrawBench prompt, generate $K = 100$ images using different initial noise seeds, and report the mean LPIPS over all pairs. In these diversity experiments, a higher LPIPS value indicates greater variation among generated images.

Statistics of the learned scale. To characterize the learned scale in Figure 6b, let $\{\sigma_d\}_{d=1}^D$ denote the standard deviations predicted by $\sigma_\varphi(c)$ after flattening the token embedding entries of a prompt, and let ρ denote the reference scale. We first quantify the overall expansion or contraction relative to the reference distribution using the mean log-ratio,

$$\mathrm{MeanLogRatio}(c) := \frac{1}{D} \sum_{d=1}^{D} \log \frac{\sigma_d}{\rho} = \log \frac{\left(\prod_{d=1}^{D} \sigma_d\right)^{1/D}}{\rho}. \tag{68}$$

Equivalently, this quantity is the logarithm of the geometric mean of the predicted standard deviations relative to the reference scale. A negative value indicates that the typical coordinate contracts below ρ, whereas a positive value indicates an overall expansion.

[1] https://github.com/microsoft/soc-fine-tuning-sd

[2] https://github.com/NVIDIA/NeMo-Aligner

Table 5: Architecture and training configurations of PTERA. The top panel reports the parameter counts of each text encoder and its associated PTERA residual module. The bottom panel summarizes the backbone training configurations. **BS**, **GS**, and **IS** denote batch size, guidance scale, and the number of inference steps, respectively.

Text encoder	Used in	Encoder params.	PTERA module params.	Overhead (%)
CLIP-L	SD1.5, SDXL, SD3	123.65M	494,848	0.400
CLIP-G	SDXL, SD3	694.66M	2,748,368	0.396
T5	SD3	4,762.31M	19,069,536	0.400

Backbone	Active encoders	PTERA params.	Learning rate	BS	(β, γ)	ρ	IS	GS
SD1.5	CLIP-L	494,848	3×10^{-4}	4	$(0.1, 0.1)$	0.1	50	7.5
SDXL	CLIP-L, CLIP-G	3,243,216	3×10^{-5}	4	$(0.1, 0.1)$	0.1	50	5.0
SD3	CLIP-L, CLIP-G, T5	22,312,752	3×10^{-7}	16	$(10^{-3}, 10^{-3})$	0.1	28	5.0

To separately quantify the directional non-uniformity of the learned scale, we define anisotropy as the log-ratio between the arithmetic and geometric means of the coordinate-wise variances,

$$\text{Anisotropy}(c) := \log \frac{\frac{1}{D}\sum_{d=1}^{D} \sigma_d^2}{\left(\prod_{d=1}^{D} \sigma_d^2\right)^{1/D}} = \log\left(\frac{1}{D}\sum_{d=1}^{D} \sigma_d^2\right) - \frac{1}{D}\sum_{d=1}^{D} \log \sigma_d^2. \tag{69}$$

By the AM-GM inequality, this quantity is nonnegative and equals zero if and only if σ_d is constant across all coordinates. Larger values therefore indicate a more anisotropic distribution, in which the variance is increasingly concentrated along a subset of embedding dimensions.

PTERA Architecture and Training Configurations. PTERA augments each frozen text encoder with a lightweight multi head attention residual module that parameterizes the mean μ_φ and scale σ_φ of the conditional text embedding distribution. For models containing multiple text encoders, we attach an encoder-specific PTERA module to each encoder rather than sharing a single module across heterogeneous embedding spaces. Accordingly, SD1.5 uses one PTERA module for CLIP-L, SDXL uses separate modules for CLIP-L and CLIP-G, and SD3 uses separate modules for CLIP-L, CLIP-G, and T5 encoders. The pre-trained text encoders and the diffusion backbone remain frozen throughout PTERA training.

Table 5 summarizes the configurations used for the PickScore steered models reported in Tables 1 and 2. Each PTERA module introduces approximately 0.4% as many trainable parameters as its corresponding frozen text encoder. The total numbers of trainable PTERA parameters are $494{,}848$, $3{,}243{,}216$, and $22{,}312{,}752$ for SD1.5, SDXL, and SD3, respectively. Then the SD3 PTERA parameter count is obtained as $494{,}848 + 2{,}748{,}368 + 19{,}069{,}536 = 22{,}312{,}752$. For a given diffusion backbone, all PTERA modules are optimized jointly using the single learning rate reported in Table 5.

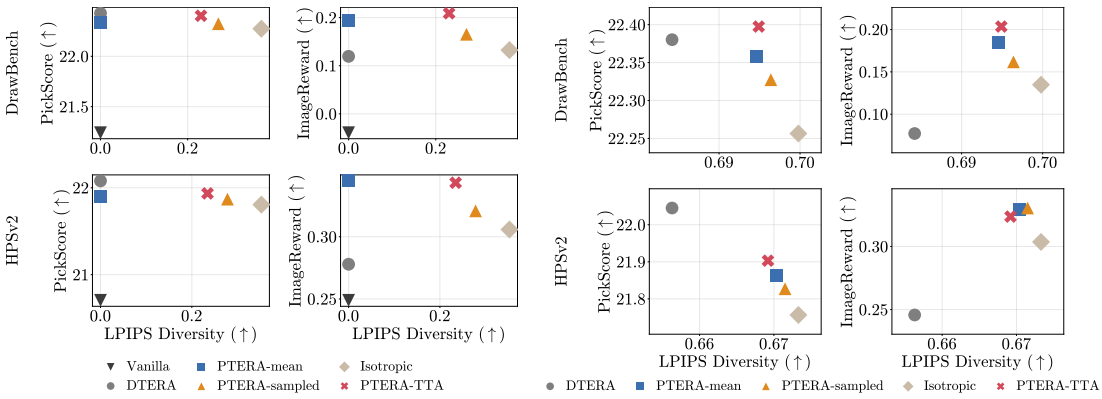
D Additional Experiment Result

D.1 Exploring reward-aware distribution of PTERA.

We extend the analysis in Figure 4b across PickScore and ImageReward, the DrawBench benchmark, the held-out HPSv2 benchmark, and two initial noise protocols. For each prompt c, we generate $K = 16$ images. In the *fixed noise* setting, all K generations start from the same initial noise x_T, thereby isolating the stochasticity only induced by text embedding sampling. In the *varied noise* setting, each generation instead starts from an independently sampled initial noise $x_T^{(k)}$.

To distinguish the directional structure learned by $\sigma_\varphi(c)$ from the overall magnitude of the perturbation, we add an isotropic baseline. Let d denote the number of scalar coordinates in the text embedding, and define

$$\bar{\sigma}_\varphi(c) := \text{RMS}(\sigma_\varphi(c)) = \frac{\|\text{vec}(\sigma_\varphi(c))\|}{\sqrt{d}}, \quad z_{\text{iso}}^{(k)} = \mu_\varphi(c) + \bar{\sigma}_\varphi(c)\varepsilon^{(k)}, \quad \varepsilon^{(k)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{70}$$



(a) Fixed initial noise x_T.

(b) Varied initial noises x_T.

Figure 7: Reward-diversity characteristics of the text embedding distributions learned by PTERA under different initial noise protocols. For each prompt, we generate 16 images and report mean pairwise LPIPS diversity . In (a), all generations for a prompt share the same initial noise x_T, isolating the stochasticity introduced through text embedding sampling. In (b), each generation starts from an independently sampled x_T, such that prior noise variability also contributes to the measured diversity. *Isotropic* preserves the PTERA mean but replaces the scale with the RMS scalar scale in Equation 70.

Figures 7a and 7b report the resulting reward-diversity trade-offs. Under fixed noise (Figure 7a), PTERA-mean is deterministic and therefore do not generate any diversity, whereas PTERA-sampled introduces substantial diversity solely through the learned text embedding distribution. Although the isotropic baseline produces greater perceptual variation, it attains lower rewards than PTERA-sampled, indicating that the learned scale captures reward relevant exploration directions rather than merely increasing the perturbation magnitude. PTERA-TTA further exploits this diversity through Algorithm 1, increasing the steering reward on average by selecting text embeddings adapted to individual denoising trajectories. When independently sampled initial noises are used (Figure 7b), the diversity induced by x_T dominates that induced in the text embedding space, and the LPIPS differences among the PTERA variants become comparatively small. Nevertheless, PTERA-TTA continues to improve the steering reward over unselected embedding samples, demonstrating that the learned distribution remains useful for reward-aware candidate selection even when sample diversity is already provided by the prior noise.

D.2 Ablation Studies.

Probabilistic conditioning. We isolate the effect of probabilistic conditioning by progressively extending a (matched parameter) deterministic text encoder while keeping the remaining training configuration fixed. **Mean-only** (*i.e.* DTERA.) conditions the diffusion model only on $z = \mu_\varphi(c)$. Its residual module is adjusted to contain 494,464 trainable parameters, closely matching the 494,848 parameters of PTERA, thereby controlling the adapter capacity.

Table 6: Ablation of the probabilistic parameterization. LPIPS Div. is computed from four images per prompt with independent initial noises.

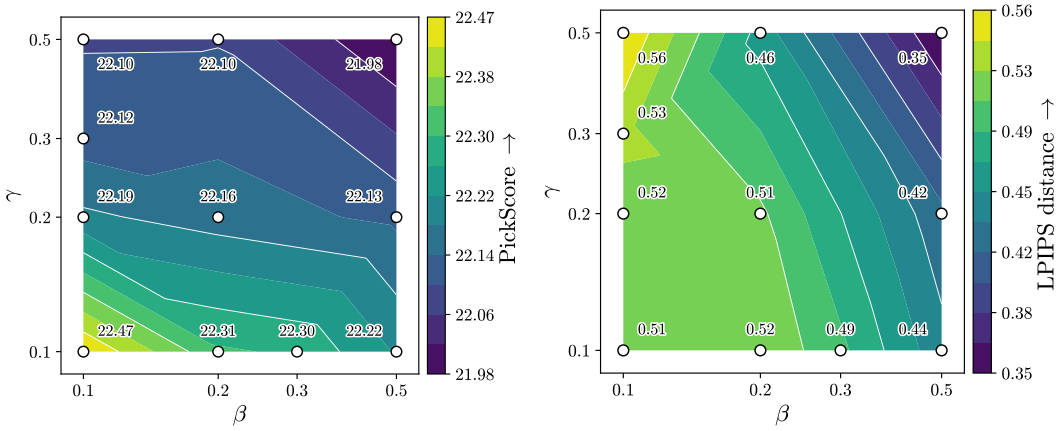
Variant	PS ↑	HPSv2 ↑	LPIPS Div. ↑
Mean-only	22.278±0.502	0.266±0.018	0.610
Fixed-Iso	22.317±0.502	**0.267**±0.019	0.616
Learned-Iso	22.339±0.554	0.265±0.018	0.618
PTERA	**22.384**±0.528	**0.267**±0.017	**0.619**

Fixed-Iso introduces stochastic conditioning with a fixed reference scale, $z = \mu_\varphi(c) + \rho\epsilon$, where $\rho = 0.1$ and $\epsilon \sim \mathcal{N}(0, I)$. **Learned-Iso** instead predicts a prompt-dependent scalar $s_\varphi(c)$ and uses $z = \mu_\varphi(c) + s_\varphi(c)\epsilon$, allowing the magnitude of exploration to adapt to each prompt while remaining isotropic. Finally, **PTERA** learns the full coordinate-wise scale, $z = \mu_\varphi(c) + \sigma_\varphi(c) \odot \epsilon$, allowing an anisotropic embedding distribution.

For evaluation, we generate four images per DrawBench prompt using four independent initial noise seeds. For each generation, the conditioning vector z is instantiated according to the corresponding variant and then used to generate the image in the stochastic variants, z is sampled from the resulting text embedding distribution. We report PickScore (PS), HPSv2, and the mean pairwise LPIPS

Table 7: Isolating the two regularizers under $\alpha = \beta + \gamma = 0.4$. LPIPS distance is measured against the vanilla SD1.5 image generated from the same prompt and initial noise. $\mathcal{D}_{\text{text}}$ and $\mathcal{D}_{\text{diff}}$ are evaluated for every model regardless of which terms were used in training. The two columns are on different scales and should be compared within a column only.

β	γ	PickScore ↑	LPIPS dist. ↓	$\mathcal{D}_{\text{text}}$ (Eq. 45)	$\mathcal{D}_{\text{diff}}$ (Eq. 40)
0.4	0.0	22.287	0.521 ± 0.124	33.00 ± 15.8	0.0079 ± 0.003
0.3	0.1	**22.297**	$\mathbf{0.494} \pm 0.118$	2.58 ± 1.1	0.0093 ± 0.004
0.2	0.2	22.164	0.512 ± 0.089	1.62 ± 0.6	0.0140 ± 0.005
0.1	0.3	22.124	0.532 ± 0.106	1.45 ± 0.6	0.0168 ± 0.007
0.0	0.4	22.285	0.569 ± 0.103	1.33 ± 0.5	0.0230 ± 0.011



(a) PickScore contour.

(b) LPIPS distance to the paired vanilla sample.

Figure 8: Independent sweep of the diffusion-level regularization coefficient β and the text-level regularization coefficient γ. White circles indicate the evaluated configurations, while the contour surfaces interpolate between the measured points. Increasing β generally produces a stronger reduction in perceptual deviation, whereas increasing γ more strongly suppresses the steering reward in the low-regularization regime.

diversity among the four images. Table 6 shows that PS and LPIPS diversity increase monotonically as the conditioning parameterization becomes more expressive. PTERA achieves the highest PickScore and diversity while matching the best HPSv2 score. These results indicate that the gain is not explained solely by adapter capacity or stochastic perturbation, but further benefits from learning the prompt-dependent geometry of the conditioning distribution.

Decomposing the regularizer, β and γ. Proposition 2 shows that $\mathcal{D}_{\text{diff}}$ and $\mathcal{D}_{\text{text}}$ are tight in complementary regimes, but it does not by itself show that both terms matter in practice. To isolate each term, we fix the total coefficient $\alpha = \beta + \gamma = 0.4$ (the larger of the two DTERA settings in Table 1) and sweep the split $(\beta, \gamma) \in \{(0.4, 0), (0.3, 0.1), (0.2, 0.2), (0.1, 0.3), (0, 0.4)\}$. The two endpoints use a single regularizer: $(0.4, 0)$ keeps only the $\mathcal{D}_{\text{diff}}$ term and $(0, 0.4)$ keeps only the $\mathcal{D}_{\text{text}}$ term.

We report four quantities. The first two are the steering reward (PickScore) and the LPIPS distance from the paired reference image (*i.e.* vanilla SD1.5 image). The last two form a cross-evaluation, for every trained model we evaluate both regularizers on the evaluation prompts with fresh noise, $\mathcal{D}_{\text{text}}$ through its closed form (Equation 45) and $\mathcal{D}_{\text{diff}}$ through Equation 40 as instantiated in training.

Table 7 shows that each regularizer, when removed, is the one whose value grows. Training with $\mathcal{D}_{\text{diff}}$ alone ($\gamma = 0$) leaves the text embedding-level divergence largely unconstrained: $\mathcal{D}_{\text{text}} = 33.0$, which is $12.8\times$ the value of the next configuration. Training with $\mathcal{D}_{\text{text}}$ alone ($\beta = 0$) leaves the image level discrepancy loosest. $\mathcal{D}_{\text{diff}} = 0.0230$, which is $2.9\times$ the value of the diffusion only configuration. Across the sweep, the means of both columns are ordered monotonically in the split.

To further examine how the two regularizers affect the learned model, we independently vary β and γ and visualize the resulting PickScore and LPIPS distance in Figure 8. Unlike the sweep under the

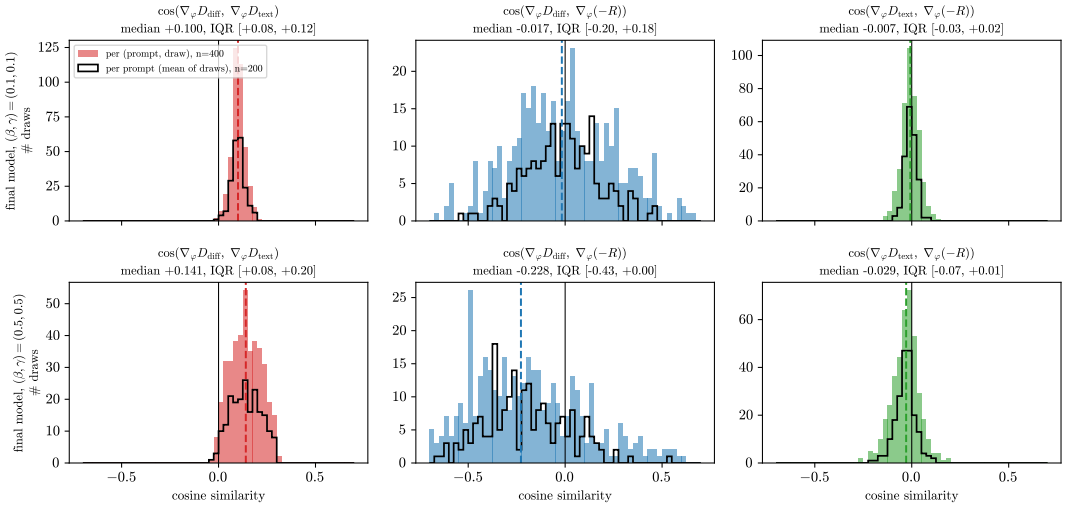


Figure 9: **The two regularizers pull the text encoder head in different directions.** For each DrawBench prompt and each draw of (x_T, ε) we replay one training forward pass of PTERA without updating parameters and back-propagate the three loss terms separately, obtaining $\nabla_\varphi D_{\text{diff}}$, $\nabla_\varphi D_{\text{text}}$ and $\nabla_\varphi(-R)$ with respect to the head parameters φ (the diffusion parameter θ is frozen). Each panel shows the cosine similarity between two of these gradients. Top row: the final model trained with $(\beta, \gamma) = (0.1, 0.1)$; bottom row: the final model trained with $(0.5, 0.5)$. **(Left.)** the two KL gradients are nearly orthogonal (median cosine 0.10), so the terms constrain largely distinct directions rather than duplicating one another. **(Middle.)** $\nabla_\varphi D_{\text{diff}}$ is on average neutral with respect to the reward at $(0.1, 0.1)$ but becomes systematically anti-aligned with it at $(0.5, 0.5)$ (median $-0.02 \rightarrow -0.23$), identifying the denoising KL as the term that directly counteracts reward ascent. **(Right.)** $\nabla_\varphi D_{\text{text}}$ is orthogonal to the reward gradient at both regularization strengths.

fixed summation value in Table 7, this experiment reveals the sensitivity of each metric to the two coefficients independently.

Surfaces of Figure 8 suggest partially distinct roles of $\mathcal{D}_{\text{diff}}$ and $\mathcal{D}_{\text{text}}$. Increasing β generally reduces LPIPS distance, indicating that $\mathcal{D}_{\text{diff}}$, which directly penalizes discrepancies between the conditional diffusion kernels, primarily controls deviation in the generated image space. In contrast, increasing γ produces a relatively strong reduction in the steering reward, particularly in the low-regularization regime. This behavior is consistent with $\mathcal{D}_{\text{text}}$ directly constraining reward-driven displacement of the learned text embedding distribution from its reference.

Importantly, these roles are not fully disentangled. Both coefficients affect both metrics, and the non-parallel contour structure indicates an interaction between the two regularizers. We therefore interpret $\mathcal{D}_{\text{diff}}$ and $\mathcal{D}_{\text{text}}$ as providing different primary forms of regularization while remaining complementary in controlling the final generated distribution.

Additionally, Figure 9 provides a complementary parameter space view of the two regularizers. Although the gradients of $\mathcal{D}_{\text{diff}}$ and $\mathcal{D}_{\text{text}}$ are weakly positively aligned, their median cosine similarities remain close to zero (0.10 and 0.14), indicating that they constrain largely distinct local directions in the PTERA head. Moreover, $\nabla_\varphi \mathcal{D}_{\text{text}}$ remains nearly orthogonal to the reward gradient, whereas $\nabla_\varphi \mathcal{D}_{\text{diff}}$ becomes increasingly anti-aligned with it under stronger regularization. Together with the coefficient sweeps above, this supports the view that $\mathcal{D}_{\text{diff}}$ and $\mathcal{D}_{\text{text}}$ play partially distinct but complementary roles rather than acting as redundant penalties.

Transferability. An additional advantage of text encoder adaptation is that the learned PTERA module can be transferred across generative models that expose an architecturally compatible conditioning branch. To examine this property, we train PTERA only on vanilla SD1.5 and apply the resulting module zero-shot to SD1.4, SD1.5-DPO (Wallace et al., 2024), and SDXL, without any additional reward optimization. As shown in Table 8, the transferred PTERA consistently improves PickScore, ImageReward, and HPSv2 over the corresponding base model across all three settings. Notably, the transfer remains effective even when the underlying diffusion model differs substantially from the SD1.5 model used during PTERA training.

For SDXL, whose conditioning stack additionally contains a CLIP-G encoder, we transfer only the residual shift learned by the SD1.5 PTERA head. Let $h_{\text{last}}(c)$ and $h_{\text{penult}}(c)$ denote the token-wise embeddings obtained from the last and penultimate hidden layers of the shared CLIP-L text encoder, respectively. We define the reward-aware shift learned by PTERA as $\Delta_\varphi(c) = \mu_\varphi(c) - h_{\text{last}}(c)$. We then condition the CLIP-L branch of SDXL with $h_{\text{penult}}(c) + \Delta_\varphi(c)$. This construction preserves CLIP-L representation of SDXL and superimposes only the reward-aware displacement learned by PTERA.

Table 8: Zero-shot transferability of PTERA trained on vanilla SD1.5. The same PTERA parameters are transferred without additional fine-tuning.

Backbone	PS ↑	IR ↑	HPSv2 ↑	AES ↑
SD1.4	21.258	-0.063	0.246	5.798
(+PTERA)	**22.141**	**0.136**	**0.262**	**5.943**
SD1.5-DPO	21.289	0.032	0.249	5.758
(+PTERA)	**22.536**	**0.322**	**0.274**	**5.965**
SDXL	22.219	0.329	0.251	5.896
(+PTERA)	**22.306**	**0.482**	**0.255**	**5.966**

This form of zero-shot transfer highlights a structural advantage of conditioning adaptation. Diffusion-parameter updates, including full fine-tuning and LoRA (Hu et al., 2022) adapters, are generally tied to the parameterization and architecture of the diffusion backbone on which they were trained and therefore cannot be directly transferred, for example, from an SD1.5 U-Net to SDXL. In contrast, PTERA can reuse its learned text-space transformation whenever a compatible conditioning interface is available, enabling reward-alignment behavior learned on one generative backbone to be reused by another.

D.3 Comparison to Another Rollout TTA methods.

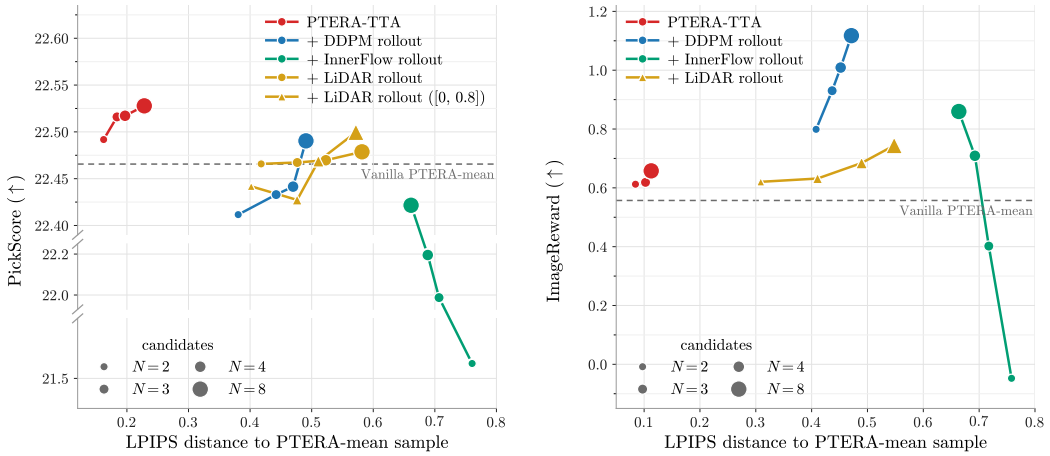
We compare PTERA-TTA with three alternative test-time alignment methods that construct reward information through stochastic exploration in the image latent space. As introduced in Section 2.3, rollout TTA methods differ primarily in how candidate future samples are constructed from an intermediate denoising state x_t. DDPM rollout (Kim et al., 2026a) draws multiple candidates $\{x_{t-1}^{(i)}\}_{i=1}^N$ from the stochastic DDPM transition kernel. Each candidate is then passed through the diffusion network to obtain a corresponding clean-sample prediction via Tweedie's formula, which is evaluated by the reward function before selecting the next state. GLASS Flows (Holderrieth et al., 2026) instead constructs an *inner flow* from the pre-trained generative model, enabling efficient sampling of stochastic transitions without retraining the backbone. We use this transition mechanism to generate rollout candidates and refer to the resulting baseline as *InnerFlow rollout*. Finally, LiDAR (Kim et al., 2026b) avoids repeatedly constructing future rollouts from each x_t. It first generates a pool of marginal lookahead samples using a few-step solver such as DPM-Solver (Lu et al., 2022), annotates them with rewards, and subsequently uses these samples to form a closed-form estimate of the reward guidance (Kim et al., 2025). Once the lookahead pool has been annotated, the target sampling stage requires neither additional reward function evaluations during inference.

PTERA-TTA takes a different approach and moves the stochastic exploration from the image latent space to the text conditioning space. At a selected timestep, PTERA-TTA keeps the current image latent x_t fixed and samples multiple conditioning vectors from the learned PTERA distribution to form the candidate clean-sample predictions. This distinction also provides a computational advantage under classifier-free guidance: because all candidates share the same $(x_t, \text{null-text embedding})$, the unconditional diffusion prediction can be reused across candidate embeddings, whereas image space rollout methods evaluate the network at different latent states. PTERA-TTA therefore reduces redundant diffusion model evaluations while performing candidate exploration in the learned reward-aware text embedding space.

For a fair comparison, all methods start from the same trained PTERA checkpoint and use its mean text embedding as the base conditioning, as summarized in Table 9. Figures 10a and 10b compare the resulting reward improvement against the LPIPS distance from the corresponding PTERA-mean sample. Under PickScore steering, the sparse selection schedule $\mathcal{S} = \{50, 40, 30, 20, 10\}$ allows PTERA-TTA to consistently improve the steering reward while remaining substantially closer to the PTERA-mean generation than the image-space rollout baselines. This behavior indicates that text space exploration provides a relatively local form of adaptation that preserves much of the perceptual content of the original generation.

Table 9: Implementation configurations and inference cost of the rollout TTA methods. For LiDAR, the values in parentheses decompose the total runtime into target sampling and lookahead construction, respectively. All runtime measurements use $N = 8$ candidates.

Model checkpoint	Rollout TTA target	Selection schedule $\mathcal{S}$	Reward weighting	Method	Sec./img ($N = 8$)
PTERA (PS)	PS	$\{50, 40, 30, 20, 10\}$	Greedy ($\alpha \to 0$)	PTERA-TTA	4.81
				DDPM rollout	5.02
				InnerFlow rollout	5.16
		$[50, 10]$	$\alpha = 2 \times 10^{-4}$	LiDAR rollout	4.74 (1.43 + 3.31)
PTERA (IR)	IR	$[40, 15]$	Greedy ($\alpha \to 0$)	PTERA-TTA	17.76
				DDPM rollout	19.44
				InnerFlow rollout	22.43
			$\alpha = 2 \times 10^{-4}$	LiDAR rollout	4.74 (1.47 + 3.27)



(a) PickScore steering.

(b) ImageReward steering.

Figure 10: Reward-deviation trade-off of PTERA-TTA and image space rollout baselines. LPIPS is measured relative to the corresponding PTERA-mean generation, such that smaller values indicate stronger preservation of the base sample. Marker size denotes the number of candidates N. **(a)** Under PickScore steering, PTERA-TTA improves the target reward while remaining in a substantially more local perceptual neighborhood. **(b)** Under ImageReward steering, image-space rollout methods can achieve larger reward gains at the cost of substantially greater perceptual deviation.

Such locality, however, can also limit the magnitude of corrective changes. When the base generation is severely misaligned with the prompt, preserving a small LPIPS distance may prevent the sampler from reaching substantially different image structures. This effect is more apparent under ImageReward steering in Figure 10b: PTERA-TTA remains closest to the PTERA-mean sample, whereas DDPM and LiDAR rollouts can attain larger reward improvements by allowing substantially greater perceptual deviation. These results suggest that PTERA-TTA provides a conservative reward-deviation trade-off rather than uniformly dominating image space rollout methods. Representative generations under PickScore steering are shown in Figure 12.

Table 9 summarizes the configurations and inference costs used in the rollout comparison. For PTERA-TTA, DDPM rollout, and InnerFlow rollout, we use the greedy best-of-N selection rule following the backward process search of Kim et al. (2026a), corresponding to the greedy selection (*i.e.* $\alpha \to 0$) of the reward weighted selection rule. We apply the same greedy rule to InnerFlow rollout so that the comparison isolates the effect of the candidate generation mechanism. LiDAR instead uses its reward weighted guidance with $\alpha = 2 \times 10^{-4}$ due to its algorithmic difference. For LiDAR, the reported runtime includes both lookahead sample construction and target sampling, with the two components shown in parentheses. All runtime measurements in the table use $N = 8$.

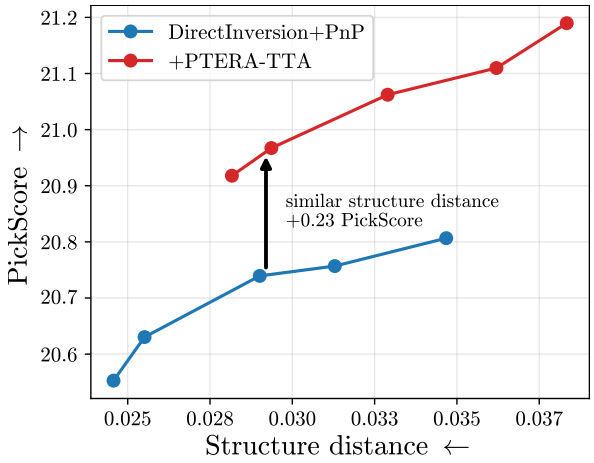


Figure 11: Alignment-preservation trade-off of PTERA-TTA on PIE-Bench with Direct Inversion and PnP editing. Each curve is obtained by sweeping the attention injection schedule of PnP (Tumanyan et al., 2023), and each point reports PickScore and structure distance averaged over the benchmark. At a matched structure distance, PTERA-TTA improves PickScore by $+0.23$.

D.4 Image Editing.

Since PTERA operates exclusively on the text conditioning pathway, it can be seamlessly plugged into downstream applications such as text-guided image editing without modifying the underlying inversion or editing mechanisms. We evaluate this plug-in capability on PIE-Bench (Ju et al., 2024), which consists of 700 source images paired with source and target prompts. Specifically, we adopt PnP editing (Tumanyan et al., 2023) with two inversion methods, DDIM inversion (Mokady et al., 2023) and Direct Inversion (Ju et al., 2024) as baselines. On top of each baseline, we replace the original text conditioning with the PTERA encoder and apply PTERA-TTA during target prompt operation. As shown in Table 10 and Figure 13, PTERA-TTA consistently improves human preference alignment metrics under both inversion pipelines, at a modest cost in source preservation. To examine whether this improvement is merely a byproduct of relaxed source preservation, we sweep the attention injection schedule of PnP, which controls the strength of source structure injection, and plot the resulting PickScore against structure distance in Figure 11. The PTERA-TTA curve lies above the baseline curve over the shared range of structure distance, yielding the reward gain at a matched structure distance, which indicates that PTERA-TTA improves the alignment preservation frontier rather than trading preservation for alignment. These results show that PTERA-TTA serves as a plug-in extension strengthening target prompt alignment in image editing, subject to a trade-off between semantic alignment and source fidelity.

We provide a brief explanation of the source preservation metrics used in Table 10, following the PIE-Bench evaluation protocol (Ju et al., 2024). **Structure distance** (Tumanyan et al., 2022) quantifies structural deviation between the source and edited images. Since self-similarity discards appearance information, this metric captures changes in spatial layout independently of texture and style. **PSNR** measures pixel-level reconstruction fidelity as the log ratio between the maximum signal power and the mean squared error. **LPIPS** (Zhang et al., 2018) measures perceptual distance as a weighted difference between deep features of a pre-trained network, which correlates better with human judgments than pixel-level metrics. **SSIM** (Wang et al., 2004) measures similarity in luminance, contrast, and local structure between the two images.

D.5 Additional Qualitative Results.

We further present qualitative image generation results for the models in Table 2 on DrawBench and PartiPrompts in Figures 14 and 15.

Table 10: Quantitative evaluation of PTERA-TTA for text-guided image editing. PTERA-TTA improves preference-based alignment metrics when applied to DDIM inversion and Direct Inversion with PnP, while exhibiting a trade-off with source preservation metrics.

Method	PickScore ↑	ImageReward ↑	HPSv2 ↑	CLIP Score ↑	Structure Distance ↓	PSNR ↑	LPIPS ↓	SSIM ↑
DDIM + PnP	21.5387	0.3217	0.2610	**25.4220**	**0.0281**	**18.9659**	**0.1930**	**0.6644**
+PTERA-TTA	**22.0381**	**0.3554**	**0.2674**	25.2969	0.0332	18.5127	0.2303	0.6359
Direct Inversion + PnP	21.6529	0.4110	0.2659	**25.4833**	**0.0241**	**19.1184**	**0.1819**	**0.6717**
+PTERA-TTA	**22.1573**	**0.4388**	**0.2719**	25.3774	0.0305	18.5965	0.2217	0.6379

E DISCUSSION

PTERA investigates whether an existing pre-trained text-to-image model can benefit from a stochastic extension of its conditioning pathway. Rather than redesigning foundation-model pretraining, we keep the pre-trained text encoder and diffusion backbone frozen and learn a reward-aware distribution over conditioning embeddings. This allows probabilistic conditioning to be introduced into existing generators without jointly retraining their representations and generative dynamics.

Under this formulation, the learned distribution should be interpreted as a reward-aware exploration distribution for a fixed generator, rather than a calibrated distribution over the semantic interpretations of a prompt. The local Gaussian reference therefore serves as a tractable stochastic extension of the deterministic encoder, rather than an intrinsic uncertainty model of the pre-trained representation.

A broader direction is to improve or learn the reference distribution itself. For example, DreamDistribution (Zhao et al., 2025) suggests that explicitly optimizing the reference text embedding distribution can substantially improve its generative utility, and such objectives could be combined with PTERA to provide a stronger stochastic reference for reward alignment. Alternatively, probabilistic representation models such as ProLIP (Chun et al., 2025), or extensions of JEPA-style representation learning (Assran et al., 2023), could provide probabilistic conditioning distributions learned from pretraining rather than introduced post hoc. Exploring how these learned reference distributions interact with PTERA-style reward adaptation is an important direction for future work.



Figure 12: Representative results under PickScore steered with various rollout TTA. All methods start from the same PTERA checkpoint (PTERA-mean). PTERA-TTA explores candidate conditioning vectors in the learned text embedding distribution, whereas DDPM, InnerFlow, and LiDAR perform reward alignment through image space sampling or guidance. PTERA-TTA generally preserves more of the PTERA-mean composition, while image space methods can induce larger structural changes.

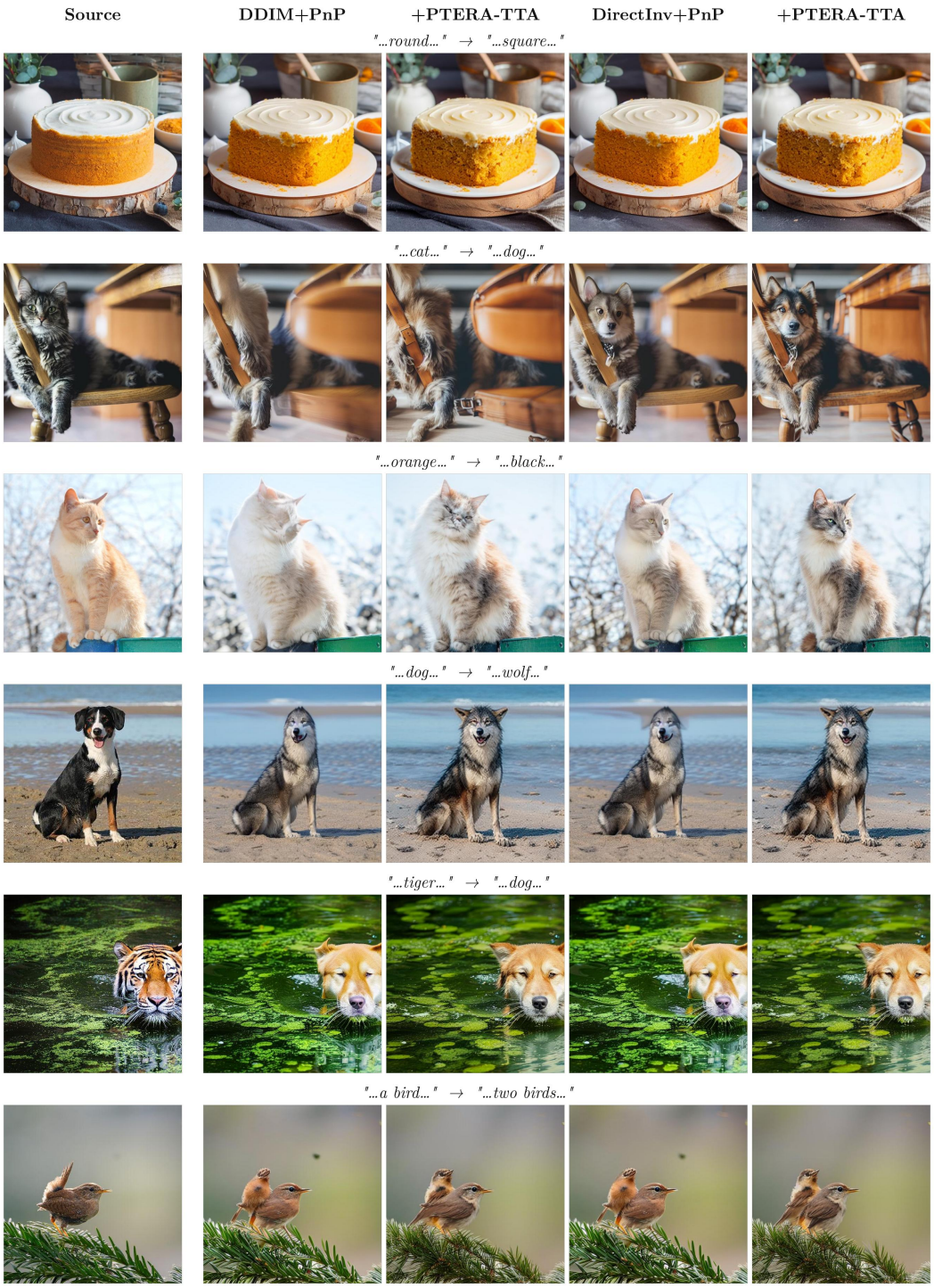


Figure 13: Qualitative results of text guided image editing based on SD1.5.



Figure 14: Qualitative results on SDXL. Top three images are randomly selected DrawBench, bottom three images are randomly selected PartiPrompts.

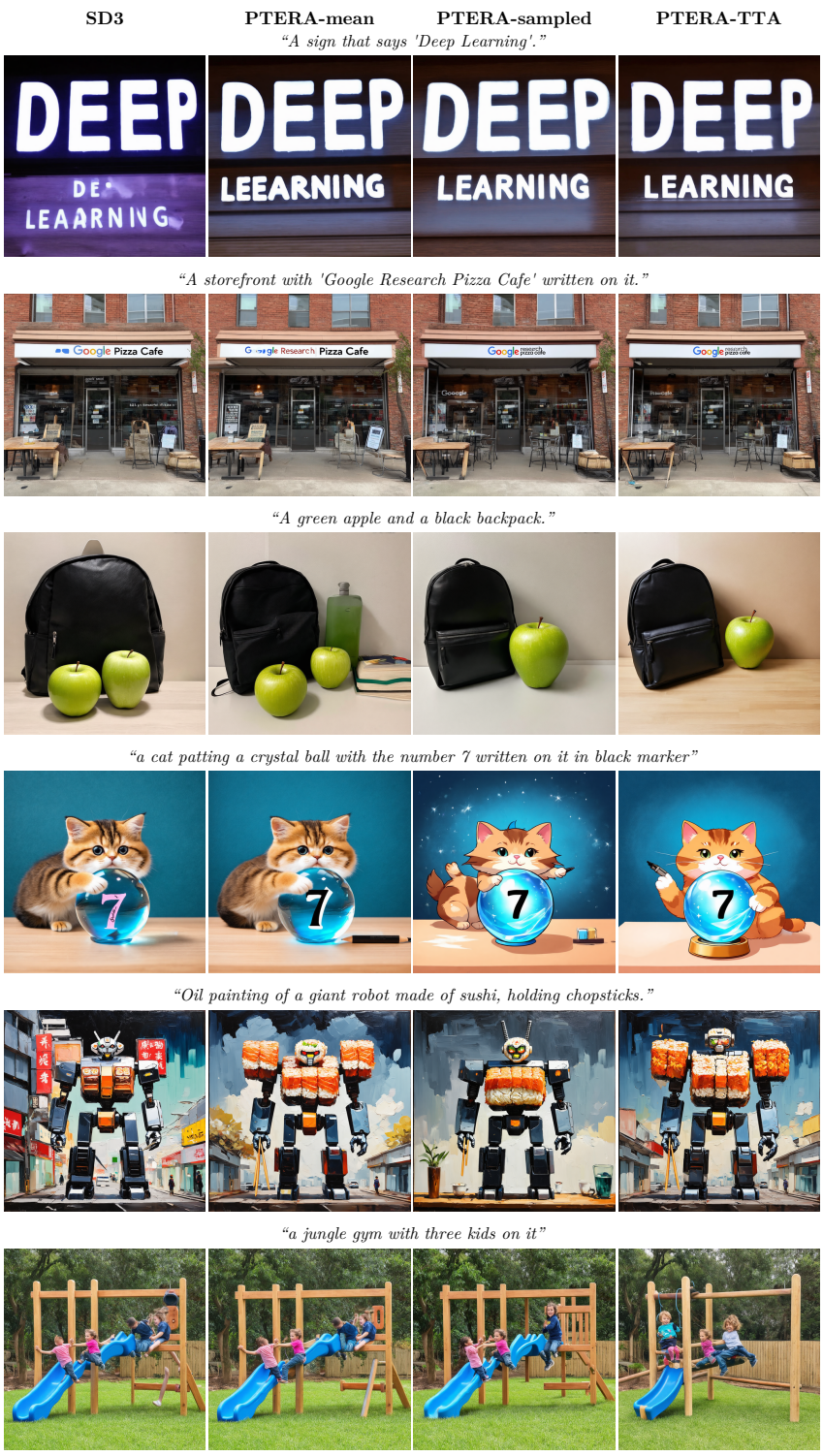


Figure 15: Qualitative results on SD3. Top three images are randomly selected DrawBench, bottom three images are randomly selected PartiPrompts.