MANIGUIDE: TRAINING-FREE PREFERENCE ALIGNMENT ACROSS DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Text-to-image generation has been advancing fast, yet the quality of a single sample still varies widely: users routinely draw many generations and cherry-pick the best one, since even a strong frozen model rarely produces its own best-case quality on a single try. Inspired by the Heeger–Bergen principle of distributional feature matching, we propose ManiGuide, a training-free inference-time guidance method that pushes a frozen generative model's typical output toward its own upper-bound quality as measured by any given metric, through patch-wise sliced-Wasserstein feature matching. Given a scoring metric, we first rank a pool of the model's own generation samples and use the top-ranked images to construct a reference manifold in deep-feature space for self-distillation. Then in generation, we identify patches whose feature distributions deviate most from the reference and guide them toward it using sliced-Wasserstein optimal transport. Across both pixel and latent image generative models, ManiGuide improves or preserves every metric. ManiGuide also generalizes beyond the base model to personalized models: on DreamBooth subject-driven fine-tunes, it turns each personalized model's own best generations into its guidance reference and improves preference metrics without additional training.

1 INTRODUCTION

Generative modeling of images has been studied for decades. Constrained by the compute of the time, early work focused on texture synthesis (Efros & Leung, 1999; Portilla & Simoncelli, 2000), where one representative approach is distribution matching (Heeger & Bergen, 1995): a convincing texture emerges by iteratively matching the per-channel histograms of a multiscale decomposition to those of a reference. As compute and large-scale data became available, learning-based methods came to dominate and the target moved from texture to natural images (Tu, 2007; Goodfellow et al., 2014; Radford et al., 2016). Diffusion models (Ho et al., 2020; Song et al., 2021; Rombach et al., 2022; Esser et al., 2024) push this further, producing images that are hard to tell apart from real ones. Sampling remains stochastic, however, so quality varies widely from one sample to the next. The common remedy is to sample repeatedly under the same condition and cherry-pick the best result (Ma et al., 2025), which is costly and wasteful.

Inspired by this principle, we propose ManiGuide, a self-distillation method (Hinton et al., 2015; Furlanello et al., 2018): the reference is built from the model's own outputs rather than an external corpus of high-quality images, and later generations are steered toward it. Given any image-quality metric, we sample a pool from the frozen model and rank it. The best- and worst-scoring samples already differ in deep-feature space, along directions the metric responds to, so we keep those directions and summarize the feature statistics of the top-ranked samples as the reference, a description of the quality the model can already reach. At sampling time ManiGuide extracts patch-level deep features from the current generation and minimizes the distributional divergence between them and that reference (Rabin et al., 2011; Bonneel et al., 2015; Heitz et al., 2021). This carries the Heeger–Bergen principle into deep-feature space, matching a full feature distribution rather than a few low-order statistics.

ManiGuide is plug-and-play across generative models and image quality metrics. We evaluate ManiGuide on the HPS v2.1 benchmark (Wu et al., 2023b), spanning two latent diffusion models (SD3.5 (Esser et al., 2024), FLUX.1-dev (Black Forest Labs, 2024)) and one pixel-space diffusion

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

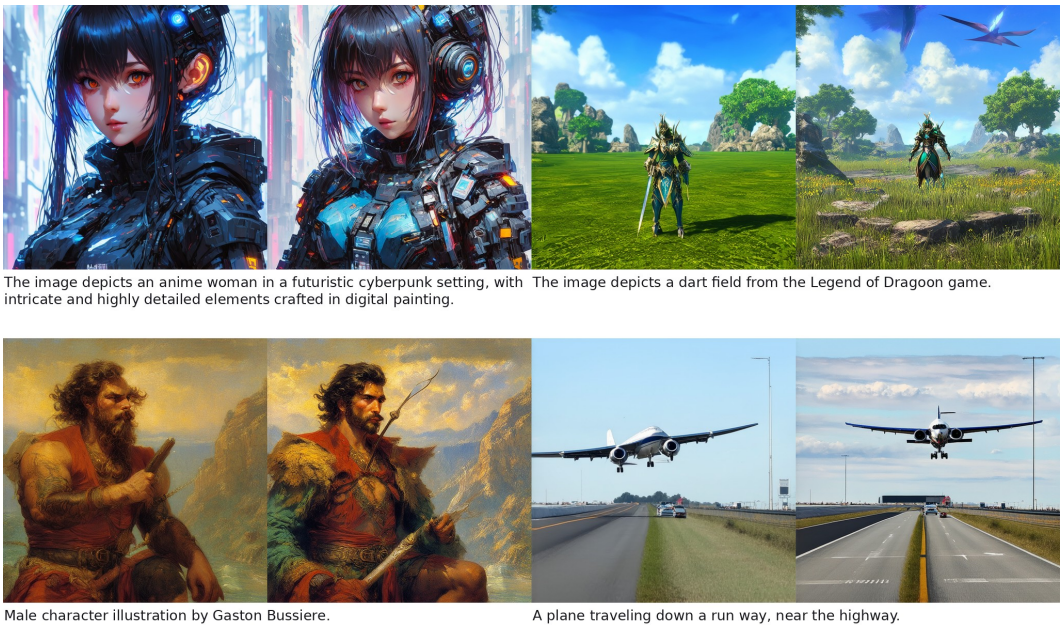

Figure 1: **Training-free, inference-time preference alignment.** Four styles in a 2×2 grid, with anime (top-left), concept-art (top-right), paintings (bottom-left), and photo (bottom-right). In each pair, the left is vanilla SD3.5-Medium and the right is guided, from the same prompt and seed.

model (PixNerd-T2I (Wang et al., 2026)) across four preference metrics (Schuhmann et al., 2022; Xu et al., 2023; Wu et al., 2023b; Kirstain et al., 2023). The results show that among different generative models, ManiGuide consistently raises or holds the targeted preference metric, while the other metrics stay non-negative, demonstrating that ManiGuide improves overall quality.

The contributions of this work are as follows:

- We propose ManiGuide, a training-free inference-time guidance method (Chung et al., 2023; Yu et al., 2023; He et al., 2024) built on self-distillation: a frozen generator is steered by the feature statistics of its own highest-ranked samples, so no external set of high-quality images is required.
- We revisit the Heeger–Bergen principle and cast guidance as distribution matching between a generation and the reference. We carry the principle from hand-crafted pixel statistics to deep features, matched with sliced-Wasserstein optimal transport (Bonneel et al., 2015; Heitz et al., 2021).
- We show that ManiGuide generalizes across generation settings, lowering FID (Heusel et al., 2017) on class-conditional ImageNet (Deng et al., 2009), improving or preserving all four preference metrics on the four-style text-to-image benchmark, and lifting eight DreamBooth (Ruiz et al., 2023) personalized models on their own subjects.

2 Related Work

Generative modeling by feature matching. Matching image statistics is one of the oldest ideas in image synthesis. Heeger & Bergen (1995) synthesize a texture by matching the histograms of its steerable-pyramid responses to those of an example, and Portilla & Simoncelli (2000) extend the match to joint wavelet statistics. Rabin et al. (2011) and Bonneel et al. (2015) cast the matching as optimal transport through Wasserstein barycenters. Gatys et al. (2016) and Ulyanov et al. (2016) then replace the hand-designed features with those of a deep network. Building on computational optimal transport (Peyré & Cuturi, 2019), Heitz et al. (2021) match such deep features with a sliced-Wasserstein loss, which we adopt.

The same principle extends from textures to natural images. Before deep learning, Tu (2007) proposes energy-based generative adversarial learning, GDL, in which a single model is both generative and discriminative. GANs (Goodfellow et al., 2014; Radford et al., 2016; Salimans et al., 2016) learn adversarial models with deep networks. GDL itself evolves into introspective neural networks (Jin et al., 2017; Lazarow et al., 2017; Lee et al., 2018), which synthesize images from a classifier and progressively retrain it to reduce the difference it can still detect between synthesized and real images. Yet Wang et al. (2025) show that a classifier still tells images of a strong diffusion model from real ones, although they look convincing, so current generators do not yet capture the full statistics of natural images. Inspired by these works, we reduce this difference at inference time by matching deep-feature statistics, without retraining the model.

Inference-time guidance and search. The variation in sample quality of a frozen diffusion model raises the question of how to steer sampling toward better images without retraining. Guidance answers it by adding a term to the model's denoising direction at each step. Dhariwal & Nichol (2021) add the gradient of a noise-aware classifier, Ho & Salimans (2022) replace the classifier with the difference between conditional and unconditional predictions, and Karras et al. (2024) guide the model with a weaker version of itself. Training-free methods (Chung et al., 2023; Song et al., 2023; Yu et al., 2023; Bansal et al., 2023) instead take the gradient of a loss on the clean-image estimate. He et al. (2024) and Yang et al. (2024) keep this update on the data manifold and the noise sphere, and Ye et al. (2024) unify the design space. Search-based methods instead select among samples, through sequential Monte Carlo and particle resampling (Wu et al., 2023a; Li et al., 2025; Kim et al., 2025; Singhal et al., 2025), best-of-N or per-step noise selection (Ma et al., 2025; Yeh et al., 2025), or initial-noise optimization (Eyring et al., 2024). These methods query a reward or a condition during generation, at every step or for every candidate. Closest to ours, Mahdizadeh Sani et al. (2026) guide toward a target distribution by matching fixed-kernel moments. We instead match the full per-patch feature distribution of the model's own best samples, and query the metric only offline to rank them.

3 METHOD

3.1 PRELIMINARIES

We build on flow-matching generative models (Liu et al., 2023; Lipman et al., 2023). At time $\sigma \in [0, 1]$ the state is $\boldsymbol{x}_\sigma = (1-\sigma)\boldsymbol{x}_0 + \sigma\boldsymbol{\epsilon}$, the model predicts the velocity $\boldsymbol{v} = g_\theta(\boldsymbol{x}_\sigma, \sigma)$ of that interpolant, and a sample is drawn by integrating it from σ=1 to σ=0 with g_θ frozen. The clean estimate $\hat{\boldsymbol{x}}_0 = \boldsymbol{x}_\sigma - \sigma\boldsymbol{v}$ is available in closed form at every step. For a latent model, $\boldsymbol{x}_\sigma$ is a latent that decodes to an image $\mathcal{D}(\boldsymbol{x}_\sigma)$ (Rombach et al., 2022; Esser et al., 2024). We use training-free guidance, which moves the state at each step along $-\nabla_{\boldsymbol{x}_\sigma}\mathcal{L}(\hat{\boldsymbol{x}}_0)$ (Yu et al., 2023; He et al., 2024).

3.2 OVERVIEW

Fig. 2 gives an overview of ManiGuide. It consists of two phases where phase 1 prepares the reference set offline and phase 2 guides the generation online.

Phase 1 (offline), once per metric. We generate a pool of images with g_θ and score each with the target metric $\mathcal{M}$. The K best- and K worst-scoring images are passed through ϕ, giving a grid of patch tokens with dimension C. We build a reference set from these tokens.

Phase 2 (online), at every generation. At each denoising step of g_θ, we estimate the clean image $\hat{\boldsymbol{x}}_0$ according to Section 3.1, extract features using ϕ, and steer the sample so its most off-reference patches move toward the Phase-1 reference (Section 3.5, Algorithm 2).

3.3 SELF-DISTILLED REFERENCE

We generate a pool of images with g_θ and score each with the target metric $\mathcal{M}$. The K best- and K worst-scoring images are passed through the feature extractor ϕ, giving high-dimensional feature tokens for each image patch; we call these the K-best and K-worst tokens. We build a reference set

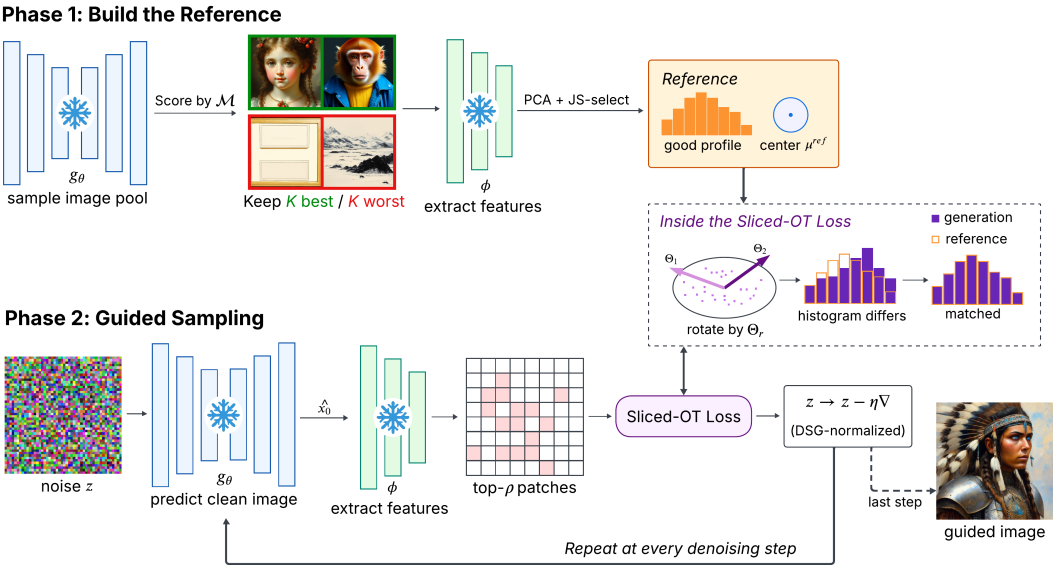

Figure 2: **Overview of ManiGuide.** Phase 1 (top) builds the reference offline. Phase 2 (bottom) guides sampling online, pushing the generation's most off-reference patches toward it.

from them, starting with the mean $\boldsymbol{\mu}_{\text{best}} \in \mathbb{R}^C$ and covariance $\boldsymbol{\Sigma}_{\text{best}} \in \mathbb{R}^{C\times C}$ of the K-best tokens,

$$\boldsymbol{\mu}_{\text{best}} = \tfrac{1}{n}\textstyle\sum_i \boldsymbol{t}_i, \qquad \boldsymbol{\Sigma}_{\text{best}} = \tfrac{1}{n-1}\Big(\textstyle\sum_i \boldsymbol{t}_i\boldsymbol{t}_i^\top - n\,\boldsymbol{\mu}_{\text{best}}\boldsymbol{\mu}_{\text{best}}^\top\Big). \tag{1}$$

and run principal component analysis on $\boldsymbol{\Sigma}_{\text{best}}$ (Jolliffe, 2002), which returns C orthonormal directions $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_C$ ordered by how much the K-best tokens vary along each.

Not every direction separates the K-best tokens from the K-worst. We histogram both sets along each direction and score that direction by the Jensen–Shannon divergence between the two histograms, keeping the m directions $\boldsymbol{u}_{j_1}, \ldots, \boldsymbol{u}_{j_m}$ with the largest divergence as the rows of $\boldsymbol{W}$:

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{u}_{j_1}^\top \\ \vdots \\ \boldsymbol{u}_{j_m}^\top \end{bmatrix} \in \mathbb{R}^{m\times C}, \qquad \boldsymbol{s} = \boldsymbol{W}\boldsymbol{t} \in \mathbb{R}^m. \tag{2}$$

The k-th entry of $\boldsymbol{s}$ is the projection $\langle \boldsymbol{u}_{j_k}, \boldsymbol{t}\rangle$, and we call $\boldsymbol{s}$ the projected token of $\boldsymbol{t}$.

Phase 1 stores three summaries of the K-best tokens. The reference center $\boldsymbol{\mu}^{\text{ref}} = \boldsymbol{W}\boldsymbol{\mu}_{\text{best}} \in \mathbb{R}^m$ is their projected mean, the quantile summary q_c records their coordinates along each kept direction c at a fixed set of percentiles, and a per-direction histogram serves only to score and select a generation's worst patches. Algorithm 1 gives the pass in full.

3.4 Distribution Matching Objective

Phase 1 provides the reference distribution. Guidance aims at distribution matching, pushing the patch token distribution of the generation to match the reference distribution. This calls for a differentiable distance between distributions. We build distances in three levels: (a) match the means, (b) match selected directions' distributions, and (c) match the tokens along every projection, and we conduct ablation studies on each level in Section 5.

Level (a): match the means. The coarsest metric is the distance between the two centroids,

$$\mathcal{L}_{\text{mean}} = \big\|\bar{\boldsymbol{s}} - \boldsymbol{\mu}^{\text{ref}}\big\|^2, \tag{3}$$

where $\bar{\boldsymbol{s}} = \frac{1}{n_p}\sum_i \boldsymbol{s}_i$ averages the generation tokens and $\boldsymbol{\mu}^{\text{ref}} = \boldsymbol{W}\boldsymbol{\mu}_{\text{best}}$ is the reference center (Section 3.3). Matching means aligns the two centers but ignores the spread.

Level (b): match selected directions' distributions. Level (a) matches only the mean of the generation tokens; level (b) matches their full distribution along each kept direction. Two one-dimensional samples are matched by rank: sort both lists and pair the i-th smallest with the i-th smallest, the optimal transport plan in one dimension, at the cost of a sort (Peyré & Cuturi, 2019). The two sides have different lengths, so we pair by percentile: the quantile summary of Section 3.3 holds the reference coordinates as $q_c[1], \dots, q_c[n_q]$ at percentiles $\alpha_k = k/(n_q{+}1)$. The generation's i-th smallest coordinate sits at percentile $\beta_i = (2i-1)/(2n_p)$ (Hyndman & Fan, 1996), and we read the summary there by interpolating between the two stored percentiles bracketing it,

$$\mathrm{interp}(\beta_i; q_c) = (1-t)\, q_c[k] + t\, q_c[k{+}1], \qquad \alpha_k \le \beta_i \le \alpha_{k+1}, \quad t = \frac{\beta_i - \alpha_k}{\alpha_{k+1} - \alpha_k}, \tag{4}$$

clamped to the end values outside $[\alpha_1, \alpha_{n_q}]$. This gives a length-n_p target, one entry per sorted coordinate. Summed over the m directions, the level-(b) loss is

$$\mathcal{L}_{1\mathrm{D}}(S; \boldsymbol{q}) = \sum_{c=1}^{m} \big\| \mathrm{sort}(s_{c,\cdot}) - \mathrm{interp}(\boldsymbol{\beta}; q_c) \big\|^2, \tag{5}$$

where $S \in \mathbb{R}^{m \times n_p}$ stacks the projected tokens being matched as columns, its c-th row $s_{c,\cdot}$ holds direction c's coordinate for every token, $\boldsymbol{q} = (q_1, \dots, q_m)$ collects the m quantile summaries, and $\mathrm{interp}(\boldsymbol{\beta}; q_c)$ applies Eq. (4) entrywise to $\boldsymbol{\beta} = (\beta_1, \dots, \beta_{n_p})$, giving the length-$n_p$ target.

Level (c): match the tokens along every projection. Level (b) constrains the marginal along each selected direction but not their joint distribution, since sorting a direction discards which token each of its coordinates came from. For example, two sets of 2D points, one along the diagonal and one along the anti-diagonal, give the same level-(b) loss yet are plainly different. Level (c) restores the joint structure by running the level-(b) match along mixed directions, each a combination of the selected directions rather than a single one. Covering more mixed directions constrains the joint distributions more tightly, and two sets that agree along every possible mixed direction are identical (Rabin et al., 2011; Bonneel et al., 2015). We draw R random rotations $\Theta_1, \dots, \Theta_R$ of the m selected directions, each supplying m orthonormal mixed ones, and average Eq. (5) over them,

$$\mathcal{L}_{\mathrm{sliced}} = \frac{1}{R} \sum_{r=1}^{R} \mathcal{L}_{1\mathrm{D}}\big(\Theta_r S; \boldsymbol{Q}_r\big). \tag{6}$$

Here $\Theta_r S$ holds the tokens' coordinates along rotation r's mixed directions, one row per direction, and $\boldsymbol{Q}_r$ collects those directions' quantile summaries, built from the K-best tokens exactly as $\boldsymbol{q}$ is. Sampling the rotations once in Phase 1 and reusing them in Phase 2 lets $\boldsymbol{Q}_r$ be precomputed (Algorithm 1). This is the sliced-Wasserstein distance (Heitz et al., 2021; Bonneel et al., 2015), the loss we minimize in all text-to-image experiments (Section 4.2).

3.5 Guided sampling

At each denoising step we extract features $\phi(\hat{\boldsymbol{x}}_0)$ from the clean estimate, producing a grid of patch tokens. Each patch is scored by its mean per-direction Jensen-Shannon divergence to the reference,

$$D_p = \frac{1}{m} \sum_{c=1}^{m} \mathrm{JS}\big(h_{p,c} \,\|\, h_c^{\mathrm{best}}\big), \tag{7}$$

where $h_{p,c}$ is patch p's coordinate-c histogram and h_c^{best} the K-best histogram computed in Phase 1. We keep the top fraction ρ of patches by D_p as a binary mask. This mask localizes the gradient to each image's most off-reference patches, leaving good patches untouched. The guidance term is the gradient $\boldsymbol{g}_0 = \partial \mathcal{L}_{\mathrm{sliced}} / \partial \boldsymbol{x}_\sigma$ of Eq. (6) evaluated on these tokens. Following Yang et al. (2024) we then project the step onto the noise sphere and rescale it,

$$\boldsymbol{g} = \sqrt{d}\, \sigma \sqrt{\rho}\, \frac{\boldsymbol{g}_0}{\|\boldsymbol{g}_0\|_2}, \qquad \boldsymbol{x}_\sigma \leftarrow \boldsymbol{x}_\sigma - \eta\, \boldsymbol{g}, \tag{8}$$

where d is the dimension of $\boldsymbol{x}_\sigma$ and $\sqrt{d}\,\sigma$ is the DSG sphere norm (Yang et al., 2024). The factor $\sqrt{\rho}$ rescales the step to account for guiding only a fraction ρ of patch locations. Algorithm 2 gives the full procedure.

Table 1: Class-conditional ImageNet generation performance at 256×256.

FID↓	PixNerd-XL	JiT-B/16	JiT-H/16
Vanilla	2.107	3.748	1.944
+ ManiGuide	**1.739**	**3.186**	**1.818**

Table 2: Human evaluation: guided win rate against the baseline (ties excluded).

Win rate (%)	Anime	C-art	Paint.	Photo	**All**
SD3.5-M	79.7	74.6	75.9	76.9	**76.8**
PixNerd-T2I	70.0	72.0	79.1	54.8	**69.4**

Table 3: Per-objective four-style benchmark (3200 prompts). Rows are the optimized target, columns the evaluated metric (IR = ImageReward, Aes = LAION aesthetic); the shaded diagonal is where the two coincide. Absolutes are in the Vanilla row, settings in Table 10.

	SD3.5-Medium				FLUX.1-dev				PixNerd-T2I			
Target	IR	Aes	HPS	Pick	IR	Aes	HPS	Pick	IR	Aes	HPS	Pick
IR	+0.072	+0.095	+0.010	+0.136	+0.041	+0.042	+0.005	+0.036	+0.016	+0.016	+0.001	+0.019
Aes	+0.044	+0.099	+0.006	+0.110	+0.026	+0.032	+0.004	+0.032	+0.011	+0.008	+0.000	+0.007
HPS	+0.062	+0.100	+0.008	+0.115	+0.004	+0.002	+0.000	+0.022	+0.016	+0.012	+0.001	+0.017
Pick	+0.061	+0.085	+0.008	+0.104	+0.037	+0.036	+0.005	+0.086	+0.016	+0.013	+0.001	+0.017
Vanilla	1.068	5.872	0.296	22.48	1.030	6.114	0.304	22.67	0.931	5.625	0.273	21.38

4 Experiments

4.1 Proof of concept: Class-conditional ImageNet

Setup. We test on class-conditional ImageNet at 256×256, measured by FID. We guide three frozen class-conditional image generation models: PixNerd-XL/P16 (Wang et al., 2026), JiT-B/16 (131M) and JiT-H/16 (953M) (Li & He, 2026). From each clean estimate $\hat{\boldsymbol{x}}_0$, we use DINOv3 encoder (Siméoni et al., 2026) to produce a 1024-dimensional feature per spatial location, and we keep the 100 directions whose real-versus-generated histograms diverge most by Jensen–Shannon divergence (Lin, 1991), analogous to Section 3.3. As the guidance objective we match the first two moments, the Level (a) mean loss of Section 3.4 plus a covariance term. See details in Section F.

Results. The guidance lowers FID (Heusel et al., 2017) on all three models as in Table 1. It therefore holds across two model families and across a sevenfold range of model size. This result suggests that the guidance serves as a plug-and-play term to improve the generator without training.

4.2 Preference alignment across four visual styles

Setup. We use HPS v2.1 (Wu et al., 2023b) for evaluation, consisting of 4 styles and 800 prompts per style. We use three text-conditional generators, two of which are latent flow-matching models, SD3.5-Medium (Esser et al., 2024) and FLUX.1-dev (Black Forest Labs, 2024), together with one pixel-space model PixNerd-T2I (Wang et al., 2026). Every image is scored with four quality metrics: ImageReward (Xu et al., 2023), the LAION aesthetic predictor (Schuhmann et al., 2022), HPS v2.1 (Wu et al., 2023b), and PickScore (Kirstain et al., 2023). For each model, we build four reference sets according to the four metrics for self-distillation (Section 3.3) on prompts disjoint from the benchmark. Every comparison is paired, with the guided and vanilla image for a prompt sharing the same initial noise and sampler. Full experiment settings are in Tables 9 and 10.

Results. Table 3 shows that for each base model and target metric, the optimized objective rises or holds, while the other three metrics stay non-negative, indicating the guidance effectively improves the overall quality instead of optimizing for the target metric at the expense of the others. In addition, the improvements hold for both latent and pixel-space models. We also provide qualitative examples for every style and both model families in Fig. 3 and Section E.

Human evaluation. We conduct a human study on one model from each regime, the latent SD3.5-Medium and the pixel-space PixNerd-T2I. We use 40 random prompts across the four styles, and 13

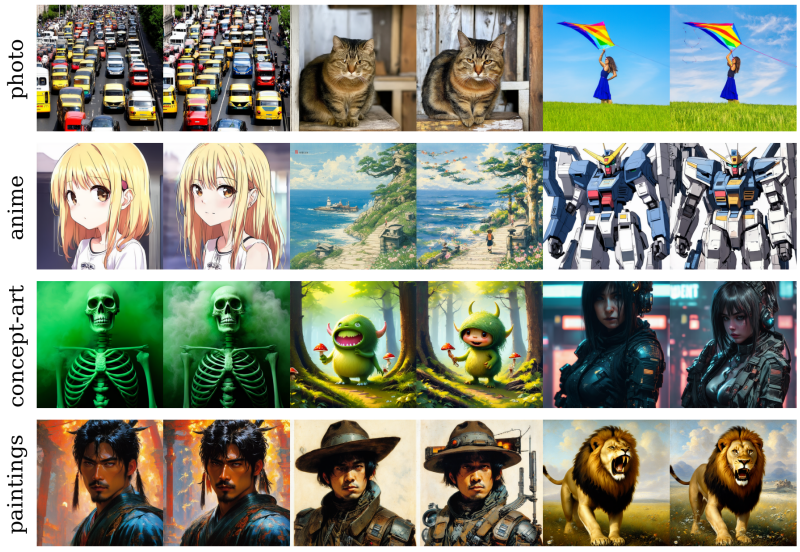

Figure 3: **What guidance changes, across all four styles.** From top to bottom, the rows show photo, anime, concept-art, and paintings; each row shows three vanilla/guided pairs, with vanilla on the left and guided on the right. Guidance sharpens surface detail and texture, while the depicted content and its agreement with the prompt are largely preserved.

Table 4: DreamBooth self-improvement on eight SDXL personalized models, on held-out trigger prompts. "van. IR" is the unmodified checkpoint's ImageReward; settings in Table 13.

	Can	Cat	Dog 5	Dog 6	Teapot	Cat 2	Dog 7	Robot	*Mean*
van. IR	0.46	0.60	0.65	0.85	0.25	0.53	0.88	0.25	—
ΔImageReward	+0.107	+0.087	+0.081	+0.074	+0.070	+0.068	+0.064	+0.050	**+0.075**
ΔAesthetic	+0.088	+0.011	+0.107	+0.059	+0.036	+0.077	+0.099	+0.068	**+0.068**
ΔPickScore	+0.165	+0.059	+0.111	+0.181	+0.033	+0.073	+0.074	+0.102	**+0.100**
ΔHPS	+0.010	+0.004	+0.005	+0.007	+0.002	+0.001	+0.000	+0.002	**+0.004**

participants compare the guided image against the baseline side by side (see details in Section G). Across both models, guided results win more than half of the decisive comparisons.

4.3 ManiGuide on personalized models

A personalized model, such as DreamBooth (Ruiz et al., 2023) or LoRA (Hu et al., 2022) fine-tune of SDXL (Podell et al., 2024), is specialized to a single appearance-defined subject. ManiGuide can be applied to such models to further improve their performance.

Setup. We train eight SDXL DreamBooth-LoRAs, one per subject of Ruiz et al. (2023), with shared official `diffusers` (von Platen et al., 2022) settings (Section D). Each frozen model is guided toward a reference self-distilled from its own ImageReward-ranked (Xu et al., 2023) samples.

Results. Every personalized model stays at or above its baseline on all four preference metrics (Table 4). ImageReward rises by a mean $+15.9\%$ ($+0.075$ absolute, up to $+0.107$ on *can*), while Aesthetic, PickScore, and HPS stay non-negative. The paired galleries are in Figs. 5 and 6.

5 Ablation study

We ablate on a controlled single-style slice, guiding SD3.5-Medium on the 800 HPS v2.1 photo prompts toward an ImageReward reference (setup in Table 11). The baseline of every table below

is the unguided generation of the same prompt with the same seed, so each number is a mean per-prompt change over a paired run; the baseline scores IR 0.902, Aes 5.295, HPS 0.273 and Pick 22.30. Negative cells are shown in red and the best per column is bold. We then ask four questions, about the reference set, the feature space, the objective, and where the guidance acts.

***Question* 1.** How to build the reference set?

Image Source. Self-distillation requires generating the reference pool, and the prompts for that generation are LAION captions, each of which comes paired with a real image. We therefore replace the self-distilled reference with the real images paired with the same 3000 captions, keeping the rest of the method fixed. The two references perform closely, and self-distillation is slightly better, ahead on three of the four metrics and tied on HPS v2.1 (Table 5). We attribute this to self-distillation being in-domain: its samples lie in the output domain of the model being guided, so the reference describes the distribution the guidance has to move, and its information can be used more directly for guidance. We further test a Per-image retrieval method, which departs from the rest of the table in that it uses different reference sets for each generated image: for each generated image it collects the C nearest real neighbors by CLIP similarity as candidates, and keeps the top 16 by ImageReward as the reference, so every image is matched against a set of its own. This tailoring buys little over self-distillation at the huge cost of a retrieval and a rebuilt reference at every generation.

Reference Size. We vary how much data the reference is built from, the size N of the pool the images are ranked from and the number K of top- and bottom-ranked images that are kept (Fig. 4a,b). The default pool size is enough, for the self-distilled pool and the real one alike, whereas the gain improves with K and then flattens. We assume a larger pool doesn't help because it pushes the selected images deeper into the tail of the scorer, which may fit the metric's bias instead of true quality. Increasing K helps and we select $K{=}16$ as the default to balance efficiency and quality.

Table 5: Where the reference images come from. The first block builds one reference and reuses it, while per-image retrieval rebuilds one for every generated image; real images are from LAION.

Reference set	ImageReward	Aesthetic	HPS v2.1	PickScore
Self-distill, 3k synthetic (ours)	+0.090	**+0.096**	+0.009	**+0.042**
Real images, 3k same-caption	+0.085	+0.088	+0.009	+0.026
Real images, 150k global pool	+0.079	+0.093	+0.009	+0.027
Per-image retrieval, $C{=}128$	+0.093	+0.091	+0.009	+0.029
Per-image retrieval, $C{=}256$	+0.089	+0.088	+0.009	+0.031
Per-image retrieval, $C{=}512$	**+0.101**	**+0.096**	**+0.010**	+0.036

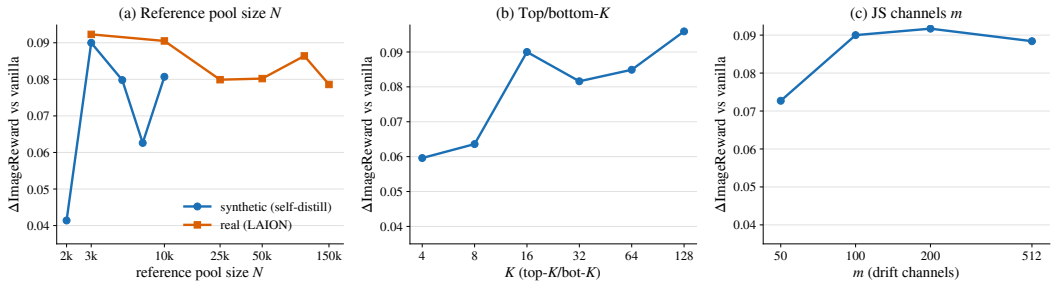

Figure 4: **Reference-side scaling is flat.** ΔImageReward versus baseline on the photo ablation slice, as a function of (a) the reference pool size N, with the synthetic pool in blue and the real pool in orange, (b) the number of retained extremes K, and (c) the number of selected directions m. Defaults are $N{=}3000$ synthetic, $K{=}16$ and $m{=}100$. The real pools are random LAION subsets and differ from the same-caption pool of Table 5, so the two are not directly comparable.

***Question* 2.** How to build the reference features?

Table 6: The feature space the reference is built in. Every row reads the same decoded image of SD3.5-Medium and changes only the encoder or the basis the m directions are selected in.

Feature space	ImageReward	Aesthetic	HPS v2.1	PickScore
SD3.5 encoder, PCA basis (ours)	$+\mathbf{0.090}$	$+0.096$	$+\mathbf{0.009}$	$+\mathbf{0.042}$
FLUX encoder, PCA basis	$+0.011$	-0.008	$+0.000$	$+0.000$
SD3.5 encoder, raw channels	$+0.060$	$+\mathbf{0.104}$	$+0.005$	$+0.019$

The reference is a distribution over feature coordinates, so it is only as good as the space those coordinates live in. Two choices define that space, the network that produces the features and the basis they are read in, and both matter. The clean estimate is decoded once with the model's own VAE decoder, and the features are read from an encoder applied to that image, so swapping the encoder changes the feature space while leaving the image untouched. Reading the same image with FLUX's encoder instead of SD3.5-Medium's almost erases the gain, from $+0.090$ to $+0.011$ ImageReward, and turns Aesthetic negative (Table 6). We interpret this as a projector tax: an encoder and a decoder trained as a pair compose to almost the identity on the model's own images, so the features are a well-conditioned readout of the latent the guidance moves, whereas a foreign encoder maps into an unrelated latent geometry and the gradient pulled back through it is misaligned. The same tax is consistent with the smaller gains of the pixel-space PixNerd-T2I in Section 4.2, which has no autoencoder of its own and must borrow one. Keeping the model's own encoder but selecting directions among the raw channels instead of the eigenbasis costs less but still a third of the gain, $+0.090$ to $+0.060$, with Aesthetic the one metric that prefers the raw channels. The number of kept directions matters least: m=100, 200 and 512 give very similar gains and only m=50 is weaker (Fig. 4c). The features must come from the model's own autoencoder, read in its eigenbasis.

***Question* 3.** Which distribution matching objective works best?

We compare three levels of loss introduced in Section 3.4. The gain grows with the strength of the match, from $+0.057$ to $+0.069$ to $+0.090$ ImageReward, and the mean match also turns HPS v2.1 and PickScore negative (Table 7). Sliced-Wasserstein is best on three of the four metrics and within noise of the 1-D form on PickScore ($+0.042$ against $+0.046$).

Table 7: Matching objective (Δ vs. baseline); all losses act on the same selected patches.

Objective	IR	Aes	HPS	Pick
Mean match	$+0.057$	$+0.004$	-0.001	-0.049
1-D OT	$+0.069$	$+0.072$	$+0.005$	$+\mathbf{0.046}$
Sliced (ours)	$+\mathbf{0.090}$	$+\mathbf{0.096}$	$+\mathbf{0.009}$	$+0.042$

Table 8: Guided fraction ρ (Δ vs. baseline). The step acts on the worst ρ of the patches.

Fraction	IR	Aes	HPS	Pick
ρ=1 (all)	-0.096	$+0.052$	$+0.009$	-0.603
ρ=0.5	$+0.087$	$+\mathbf{0.122}$	$+\mathbf{0.015}$	-0.128
ρ=0.25 (ours)	$+\mathbf{0.090}$	$+0.096$	$+0.009$	$+\mathbf{0.042}$

***Question* 4.** Which patch tokens should the guidance apply on?

Not all patches need guidance, so we vary the fraction ρ of the worst-scoring patch tokens the guidance step is restricted to, from 0.25 to 1. Guiding every token (ρ=1) degrades the ones that needed no correction, dropping ImageReward by 0.096 and PickScore by 0.603 (Table 8). Halving the mask to ρ=0.5 recovers most of that loss but still leaves PickScore negative, and only ρ=0.25 keeps all four metrics positive. Aesthetic and HPS v2.1 peak at ρ=0.5 instead, the one place where a wider mask helps. The guidance should therefore act on a small, worst-scoring fraction of tokens.

6 Conclusion

We presented ManiGuide, a training-free guidance method that boosts generation quality via distribution matching between the model's own samples and a reference set. ManiGuide is plug-and-play across models, metrics, and tasks: class- and text-conditional and personalized generation.

Ethics Statement

The human study collects only pairwise image preferences, from raters who opted in (Section G). All models and datasets we use are publicly available.

Reproducibility Statement

Everything needed to reproduce our results is included in the paper and its appendix. Section 3 defines the reference construction and the guidance objective, and Section A gives both phases as pseudocode (Algorithms 1 and 2). Section B lists every hyperparameter, the per-model guidance settings and the paired generation seeds (Tables 9 to 11). The evaluation prompts, the personalized-model training, the class-conditional setup and the human study are specified in Sections C, D, F and G. All generators, preference metrics and datasets we use are publicly available.

AI Use Statement

AI tools were used only to assist with visualizing results and to make minor refinements to language and grammar. The authors reviewed the AI-assisted content and take responsibility for the final text, figures, and scientific claims.

References

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 843–852, 2023. doi: 10.1109/cvprw59228.2023.00091.

Black Forest Labs. FLUX.1. https://github.com/black-forest-labs/flux, 2024.

Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon Wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015. doi: 10.1007/s10851-014-0506-3.

Hyungjin Chung, Jeongsol Kim, Michael T. McCann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations (ICLR)*, 2023.

Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, 2nd edition, 1988.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009. doi: 10.1109/cvpr.2009.5206848.

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 8-bit optimizers via block-wise quantization. In *International Conference on Learning Representations (ICLR)*, 2022.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 8780–8794, 2021.

Gintare Karolina Dziugaite, Daniel M. Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2015.

Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the Seventh IEEE International Conference on Computer Vision (ICCV)*, volume 2, pp. 1033–1038. IEEE, 1999. doi: 10.1109/ICCV.1999.790383.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235

of *Proceedings of Machine Learning Research*, pp. 12606–12633. PMLR, 2024. URL https://proceedings.mlr.press/v235/esser24a.html.

Luca Eyring, Shyamgopal Karthik, Karsten Roth, Alexey Dosovitskiy, and Zeynep Akata. ReNO: Enhancing one-step text-to-image models through reward-based noise optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 125487–125519, 2024. doi: 10.52202/079017-3987.

Tommaso Furlanello, Zachary C. Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International Conference on Machine Learning (ICML)*, volume 80, pp. 1607–1616, 2018.

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2414–2423, 2016. doi: 10.1109/CVPR.2016.265.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 27, 2014.

Yutong He, Naoki Murata, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Dongjun Kim, Wei-Hsiang Liao, Yuki Mitsufuji, J. Zico Kolter, Ruslan Salakhutdinov, and Stefano Ermon. Manifold preserving guided diffusion. In *International Conference on Learning Representations (ICLR)*, 2024.

David J. Heeger and James R. Bergen. Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pp. 229–238, 1995. doi: 10.1145/218380.218446.

Eric Heitz, Kenneth Vanhoey, Thomas Chambon, and Laurent Belcour. A sliced Wasserstein loss for neural texture synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9412–9420, 2021.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7514–7528, 2021. doi: 10.18653/v1/2021.emnlp-main.595.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020.

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 *ICLR*, 2022.

Rob J. Hyndman and Yanan Fan. Sample quantiles in statistical packages. *The American Statistician*, 50(4):361–365, 1996. doi: 10.1080/00031305.1996.10473566.

Long Jin, Justin Lazarow, and Zhuowen Tu. Introspective classification with convolutional nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

I. T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer-Verlag, New York, 2 edition, 2002. doi: 10.1007/b98835.

Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 52996–53021, 2024. doi: 10.52202/079017-1679.

Sunwoo Kim, Minkyu Kim, and Dongmin Park. Test-time alignment of diffusion models without reward over-optimization. In *International Conference on Learning Representations (ICLR)*, 2025.

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 (NeurIPS)*, volume 36, pp. 36652–36663, 2023. doi: 10.52202/075280-1594.

Justin Lazarow, Long Jin, and Zhuowen Tu. Introspective neural networks for generative modeling. In *ICCV*, pp. 2774–2783, 2017.

Kwonjoon Lee, Weijian Xu, Fan Fan, and Zhuowen Tu. Wasserstein introspective neural networks. In *CVPR*, pp. 3702–3711, 2018.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 36115–36125, 2026.

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gokcen 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 *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 105884–105922, 2025.

Yujia Li, Kevin Swersky, and Richard Zemel. Generative moment matching networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 37, pp. 1718–1727, 2015.

Jianhua Lin. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991. doi: 10.1109/18.61115.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, and Saining Xie. Scaling inference time compute for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2523–2534, 2025.

Matina Mahdizadeh Sani, Nima Jamali, Mohammad Jalali, and Farzan Farnia. MMD guidance: Training-free distribution adaptation for diffusion models via maximum mean discrepancy guidance. In *International Conference on Machine Learning (ICML)*, 2026. URL https://openreview.net/forum?id=WTgOxAio8a.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research (TMLR)*, 2024. URL https://openreview.net/forum?id=a68SUt6zFt.

Gabriel Peyré and Marco Cuturi. Computational optimal transport with applications to data sciences. *Foundations and Trends in Machine Learning*, 11(5–6):355–607, 2019. doi: 10.1561/2200000073.

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 (ICLR)*, 2024.

Javier Portilla and Eero P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1):49–70, 2000. doi: 10.1023/a:1026553619983.

Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein barycenter and its application to texture mixing. In *Scale Space and Variational Methods in Computer Vision (SSVM)*, volume 6667 of *Lecture Notes in Computer Science*, pp. 435–446. Springer, 2011. doi: 10.1007/978-3-642-24785-9_37.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2016.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, volume 139, pp. 8748–8763, 2021.

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

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 *CVPR*, pp. 22500–22510, 2023.

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 (NeurIPS)*, volume 35, pp. 36479–36494, 2022. doi: 10.52202/068431-2643.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, 2016.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, volume 35, pp. 25278–25294, 2022. doi: 10.52202/068431-1833.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seung Eun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *Transactions on Machine Learning Research (TMLR)*, 2026. URL https://openreview.net/forum?id=2NlGyqNjns.

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. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 267, pp. 55810–55827, 2025.

Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations (ICLR)*, 2023.

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 (ICLR)*, 2021.

Zhuowen Tu. Learning generative models via discriminative approaches. In *CVPR*, pp. 1–8, 2007. doi: 10.1109/cvpr.2007.383035.

Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 48, pp. 1349–1357, 2016.

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

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

Haowen Wang, Guowei Zhang, Xiang Zhang, Zeyuan Chen, Haiyang Xu, Dou Hoon Kwark, and Zhuowen Tu. Exploring the equivalence of closed-set generative and real data augmentation in image classification. *arXiv preprint arXiv:2508.09550*, 2025.

Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. PixNerd: Pixel neural field diffusion. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=BDnOrExHmt.

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

Luhuan Wu, Brian L. Trippe, Christian A. Naesseth, David M. Blei, and John P. Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 31372–31403, 2023a. doi: 10.52202/075280-1363.

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*, 2023b.

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 (NeurIPS)*, volume 36, pp. 15903–15935, 2023. doi: 10.52202/075280-0700.

Lingxiao Yang, Shutong Ding, Yifan Cai, Jingyi Yu, Jingya Wang, and Ye Shi. Guidance with spherical Gaussian constraint for conditional diffusion. In *International Conference on Machine Learning (ICML)*, volume 235, pp. 56071–56095, 2024.

Haotian Ye, Haowei Lin, Jiaqi Han, Minkai Xu, Sheng Liu, Yitao Liang, Jianzhu Ma, James Zou, and Stefano Ermon. TFG: Unified training-free guidance for diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 22370–22417, 2024. doi: 10.52202/079017-0704.

Po-Hung Yeh, Kuang-Huei Lee, and Jun-cheng Chen. Training-free diffusion model alignment with sampling demons. In *International Conference on Learning Representations (ICLR)*, 2025.

Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. FreeDoM: Training-free energy-guided conditional diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23174–23184, 2023.

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 (CVPR)*, pp. 586–595, 2018.

A ALGORITHMS

The two phases of Section 3 are collected here. Algorithm 1 is Phase 1, which builds the self-distilled reference once before sampling (Section 3.3), and Algorithm 2 is Phase 2, the guided-sampling loop run at every generation (Section 3.5).

Algorithm 1 Reference construction (run once, offline, for the chosen metric)

Require: frozen sampler g_θ, feature map ϕ, target metric $\mathcal{M}$
Require: sizes: K images kept per side, m directions, R rotations, n_q quantiles

Step 1: rank the model's own outputs (the only place $\mathcal{M}$ is used).
generate a pool of images with g_θ
score every image with $\mathcal{M}$ ▷ $\mathcal{M}$ only sorts; may be non-differentiable
BEST $\leftarrow$ K highest-scoring images; WORST $\leftarrow$ K lowest-scoring

Step 2: turn the kept images into feature tokens.
$X_{\text{BEST}} \leftarrow$ tokens of $\phi(x)$ for all $x \in$ BEST ▷ one C-vector per location
$X_{\text{WORST}} \leftarrow$ tokens of $\phi(x)$ for all $x \in$ WORST

Step 3: find the directions where the two sets disagree.
$\boldsymbol{\mu}_{\text{best}}, \boldsymbol{\Sigma}_{\text{best}} \leftarrow$ mean and covariance of X_{BEST} ▷ Eq. (1)
$\boldsymbol{U}, \boldsymbol{\Lambda} \leftarrow \text{eig}(\boldsymbol{\Sigma}_{\text{best}})$ ▷ PCA: directions of the K-best tokens
score each direction by $\text{JS}\big(\text{hist}(X_{\text{BEST}}) \,\|\, \text{hist}(X_{\text{WORST}})\big)$
$\boldsymbol{W} \leftarrow$ the m directions of largest JS ▷ the quality-discriminative directions

Step 4: summarize the K-best distribution on those directions.
keep mean $\boldsymbol{\mu}^{\text{ref}}$ and per-direction histograms of $\boldsymbol{W} X_{\text{BEST}}$
for $r = 1, \ldots, R$ **do** ▷ sliced-OT targets: marginals along random rotations
  draw a random rotation Θ_r
  $\boldsymbol{Q}_r \leftarrow$ sorted n_q quantiles of each rotated coordinate of $\Theta_r \, \boldsymbol{W} X_{\text{BEST}}$
end for

return reference $\big\{ \boldsymbol{W},\ \boldsymbol{\mu}^{\text{ref}},\ \text{histograms},\ \{\Theta_r, \boldsymbol{Q}_r\}_{r=1}^{R} \big\}$

Quantile summaries and percentile matching. For a kept direction c, the reference coordinates are the K-best tokens projected onto direction c (one scalar per token). The quantile summary of Section 3.3 stores their quantiles at n_q evenly spaced levels $\alpha_k = k/(n_q{+}1)$, $k = 1, \ldots, n_q$. Each $q_c[k]$ is the value below which a fraction α_k of those coordinates fall. To compare n_p generation tokens to this summary we sort their c-coordinates, place the i-th smallest at level $\beta_i = (2i - 1)/(2n_p)$ (Hyndman & Fan, 1996), and read the summary there by the interpolation of Eq. (4). This gives a length-n_p target list $\widehat{q}_c$ for any n_p, so a small token set and a large reference are matched at the same percentiles. With $n_q = 128$ and n_p in the thousands (Table 9), the stored summary is two orders of magnitude smaller than the coordinates it stands for, and its cost is independent of the reference size KN. Under this convention Eq. (5) is a consistent estimate of the squared 1-D Wasserstein-2 distance $W_2^2 = \int_0^1 (F_{\text{ref},c}^{-1}(u) - F_{S,c}^{-1}(u))^2 \, du$ (Peyré & Cuturi, 2019) between the patch and the reference along direction c.

B HYPERPARAMETERS AND IMPLEMENTATION

All hyperparameter values are collected here, using the symbols defined at first use in Section 3. All experiments run on a single server with 8× NVIDIA A6000 GPUs.

Algorithm 2 Self-distilled OT guidance (run online while generating each image)

Require: frozen sampler g_θ and feature map ϕ; reference from Algorithm 1
Require: strength η, guided fraction ρ

$\boldsymbol{x}_\sigma \leftarrow$ initial noise
for each denoising step (noise level σ) **do** ▷ guidance applied at every step
  $\boldsymbol{v} \leftarrow g_\theta(\boldsymbol{x}_\sigma, \sigma, \text{prompt})$ ▷ one model evaluation
  $\hat{\boldsymbol{x}}_0 \leftarrow \boldsymbol{x}_\sigma - \sigma \boldsymbol{v}$ ▷ pseudo-clean image estimate ($\boldsymbol{v}$ detached)
  $\boldsymbol{s} \leftarrow \boldsymbol{W}\,\phi(\hat{\boldsymbol{x}}_0)$ ▷ project the current estimate onto the selected directions
  split $\boldsymbol{s}$ into patches; keep the top-ρ most off-reference patches
  $\mathcal{L} \leftarrow$ sliced-OT distance of the kept patches to the reference ▷ Eq. (6)
  $\boldsymbol{g} \leftarrow \partial\mathcal{L}/\partial\boldsymbol{x}_\sigma$ ▷ gradient through frozen ϕ only
  rescale $\boldsymbol{g}$ onto the noise sphere ▷ Eq. (8)
  $\boldsymbol{x}_\sigma \leftarrow \boldsymbol{x}_\sigma - \eta\,\boldsymbol{g}$ ▷ step the sample toward the reference
  $\boldsymbol{x}_\sigma \leftarrow \mathrm{scheduler_step}(g_\theta, \boldsymbol{x}_\sigma, \sigma)$ ▷ ordinary denoising step
end for

return image decoded from $\boldsymbol{x}_\sigma$

Table 9: Shared settings (top) and per-model base-sampler settings (bottom) for SD3.5-Medium and FLUX.1-dev. Per-model guidance settings (η, ρ) are in Table 10; PixNerd-T2I reuses the shared settings above through a borrowed SD3.5-Medium VAE.

Setting	SD3.5-Medium	FLUX.1-dev
feature map ϕ	VAE mid-encoder (512 ch)	
feature grid $H \times W$	128×128	
selected directions m (of C)	100 (of 512)	
reference size K (best / worst)	16 / 16	
patch size $P \times P$ (grid $G \times G$)	8×8 (16×16)	
rotations R, quantiles n_q	8, 128	
token subsample for statistics	60,000	
histogram bins	64	
per-patch covariance term	none	
guidance schedule	every denoising step	
batch size	1	
generation seed (paired)	$42{,}000{,}000 + i$ (image index i)	
base sampler steps	40	50
classifier-free guidance	4.5	3.5
image resolution	512×512	

PixNerd-T2I guidance path. PixNerd-T2I generates pixels directly and has no autoencoder of its own. Its guidance therefore encodes each clean estimate into SD3.5-Medium's latent space, takes the guidance step there, and returns the correction to pixels as the difference of the two decodings (Section K). Its base sampler runs 25 steps at classifier-free guidance 4.0 and 512×512 resolution, identical for the vanilla and guided runs.

Table 10: Per-model guidance settings (η, ρ) used for Table 3. Each is shared across a model's four objectives and across the four styles; only the reference changes per objective. The lone within-model exception is the patch fraction for FLUX HPS v2.1.

Base model	strength η	patch fraction ρ
SD3.5-Medium	0.008	0.25
FLUX.1-dev	0.016	0.25 (0.10 for HPS v2.1)
PixNerd-T2I	0.01	0.10

Table 11: Setup for the ablation study of Section 5 (Table 8).

Setting	Value
base model	SD3.5-Medium
evaluation prompts	800 HPS v2.1 photo
reference pool	3000 photo captions (“, photo” suffix)
ranking objective	ImageReward
patch fraction ρ	0.25
strength η	0.008
other settings	shared defaults (Table 9)

Reference details. The K-best and K-worst sets are the $K{=}16$ highest- and lowest-scoring generations (by the target metric) from an independent self-generated pool. The base model produces that pool from the first 1000 LAION-Aesthetics (Schuhmann et al., 2022) captions per style, each appended with that style's suffix (4000 captions in total), a prompt source disjoint from the 800-per-style evaluation set. The single-style photo ablation of Table 11 instead uses a larger 3000-caption photo-only pool. For efficiency we estimate the K-best statistics from a uniform random subsample of the feature-map tokens, keeping all C channels but only a subset of the $H \times W$ spatial locations (Table 9). The covariance Eq. (1) and the sliced quantiles are computed on this subsample, and both direction selection and patch scoring use shared-edge histograms with out-of-range tokens clamped into the edge bins. The guidance gradient uses the $\hat{\boldsymbol{x}}_0$-shortcut (detached velocity) and the normalized update Eq. (8); patch selection is recomputed every guided step. The reference needs only a handful of the model's own best generations: the gain rises sharply from $K{=}8$ to $K{=}16$ and then saturates (Fig. 4b), so we use $K{=}16$ and the method needs no large curated dataset.

C Evaluation prompts

We evaluate on the four-style HPS v2.1 benchmark (Wu et al., 2023b): 800 prompts each for anime, concept-art, paintings, and photo (3200 total). The evaluation prompts are used verbatim, with no style suffix appended; the per-style suffix (e.g. `, anime`) is added only to the LAION captions that build the reference pool (Section B). Three example evaluation prompts per style follow; the full set is the public HPS v2.1 benchmark (Wu et al., 2023b).

- **anime**: “Spongebob depicted in the style of Dragon Ball Z.”; “A white-haired girl in a pink sweater looks out a window in her bedroom.”; “A hyper-realistic representation of the hypnotoad from Futurama.”
- **concept-art**: “Product still of the new iPhone 2.0 in 2029.”; “Elves attacking New York City.”; “A black wolf standing on a fallen tree in a winter forest.”
- **paintings**: “Museum painting of a mouse stealing cheese artwork.”; “A symmetrical oil painting of two waterfalls in a dense forest.”; “A detailed ink illustration of a hedgehog.”
- **photo**: “A man taking a drink from a water fountain.”; “A smiling man is cooking in his kitchen.”; “A beautiful blue and pink sky overlooking the beach.”

D Personalized-model training

Each personalized model is one SDXL DreamBooth-LoRA, trained with the official `diffusers` (von Platen et al., 2022) settings and identical for every subject (Table 12). We keep prior-preservation on, which preserves class semantics and prevents language drift (Ruiz et al., 2023, §3). The per-subject guidance hyperparameters are in Table 13.

E Qualitative examples

DreamBooth self-improvement gallery. Figures 5 and 6 show six selected recontextualization prompts for each of the eight personalized models of Table 4 (four per figure), at each subject's (η, ρ) of Table 13. The improvement is consistent across subject types (rigid objects, toys, and

Table 12: DreamBooth-LoRA training hyperparameters, identical for every subject.

Setting	Value
base model	SDXL (Podell et al., 2024)
adapter	rank-32 UNet LoRA
training steps	1000
resolution	1024×1024
learning rate	10^{-4}
batch size	1
optimizer	8-bit Adam (Dettmers et al., 2022) (bf16)
rare identifier	"sks"
prior preservation	on (λ_{pr}=1, 100 class images/subject)
training seed	0

Table 13: Per-subject guidance hyperparameters (η, ρ) for the eight personalized models of Table 4 (ImageReward objective); all other guidance settings follow Table 9, and sampling uses SDXL's default settings.

Subject	η	ρ	Subject	η	ρ
Can	0.004	0.25	Teapot	0.004	0.25
Cat	0.012	0.25	Cat 2	0.008	0.25
Dog 5	0.008	0.25	Robot	0.012	0.25
Dog 6	0.008	0.25	Dog 7	0.008	0.25

live animals) and is appearance-level, with sharper materials and lighting while the depicted subject stays recognizable and on-prompt.

Text-to-image generation examples. Figures 7 to 10 show ten examples per style for SD3.5-Medium, one figure per style. Examples are drawn from the four-style HPS v2.1 benchmark of Section 4.2 and guided toward the ImageReward reference. The examples are chosen by the authors. They show cleaner materials and lighting, with the depicted content preserved.

Pixel-space (PixNerd-T2I) examples. Figures 11 to 14 show PixNerd-T2I examples for each style, guided at the ImageReward objective with the settings of Table 10. Each figure shows four pairs, chosen by the authors from among each style's hundred largest aesthetic improvements, where the gains of the ImageReward-guided run are most visible.

F Class-conditional ImageNet pilot

This appendix details the class-conditional proof of concept of Section 4.1, the simplest instantiation of the idea that motivated the full method.

Setup. We guide three pixel-space class-conditional models on ImageNet (Deng et al., 2009) at 256×256: **PixNerd-XL/P16** (Wang et al., 2026) and two sizes of **JiT** (Li & He, 2026), JiT-B/16 (131M) and JiT-H/16 (953M). We follow the ADM evaluation protocol (Dhariwal & Nichol, 2021), scoring 50k class-balanced samples with FID (Heusel et al., 2017) against its reference statistics. Vanilla and guided generations share the class and the initial noise at each index, so the comparison is paired.

Reference and selected directions. A frozen DINOv3-L/16 encoder (Siméoni et al., 2026) produces a 1024-channel feature grid from the clean estimate $\hat{x}_0$ (summed over several encoder layers). Analogous to Section 3.3, we project out the leading content principal components and keep the 100 directions whose real-versus-generated histograms diverge most (Jensen–Shannon). The real-side moments $\boldsymbol{\mu}^{\text{ref}}, \boldsymbol{\Sigma}^{\text{ref}}$ are computed once from the real 50k ImageNet reference pool.

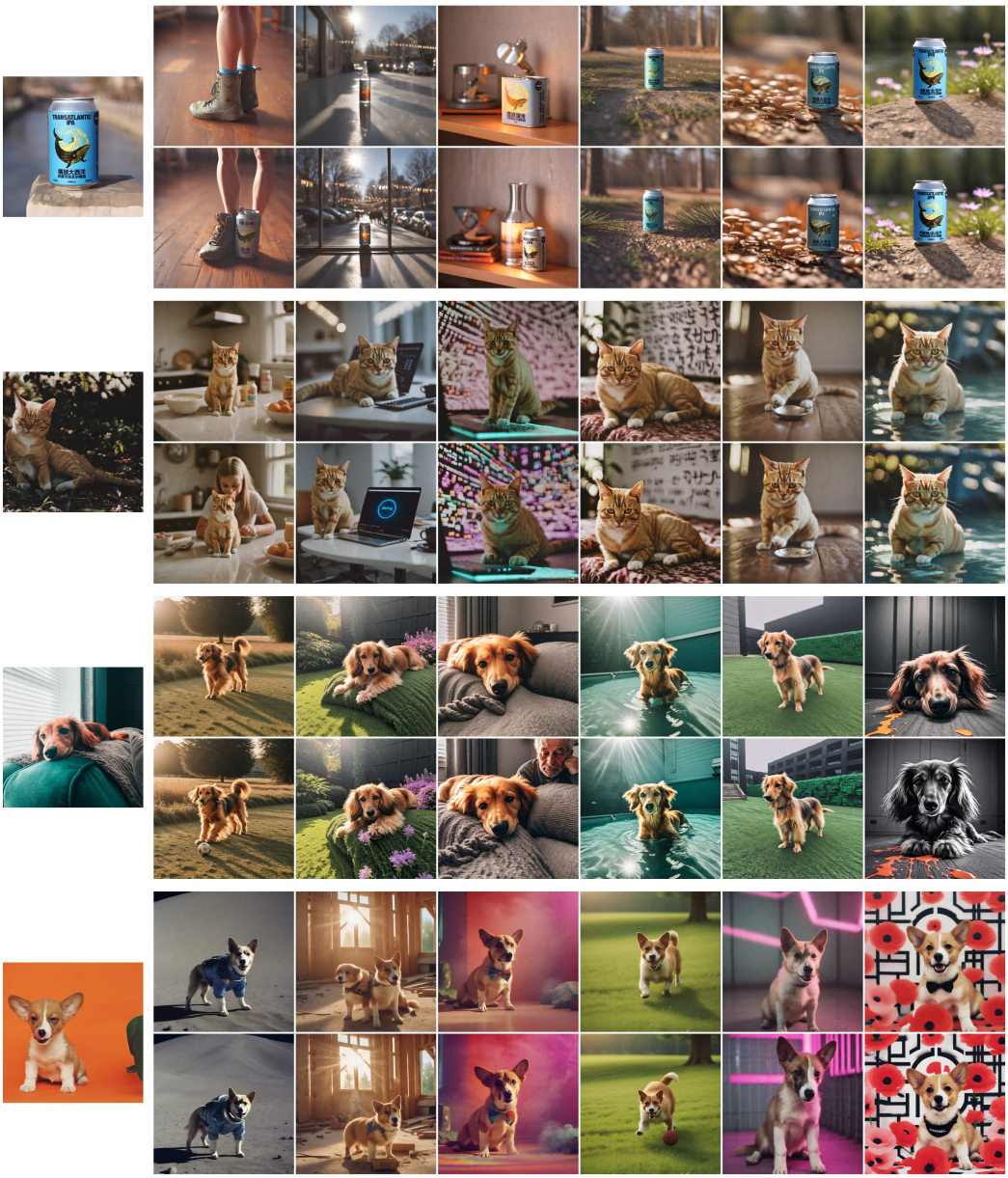

Figure 5: **DreamBooth self-improvement, full gallery (1 of 2), vanilla over guided.** Six selected recontextualization prompts per personalized model; the leftmost column is a DreamBooth training (reference) image of the subject, and within each demo the top image is vanilla and the bottom is our guided result (same prompt and seed), at the guidance settings of Table 13. Rows from top to bottom are can, cat, dog 5, dog 6.

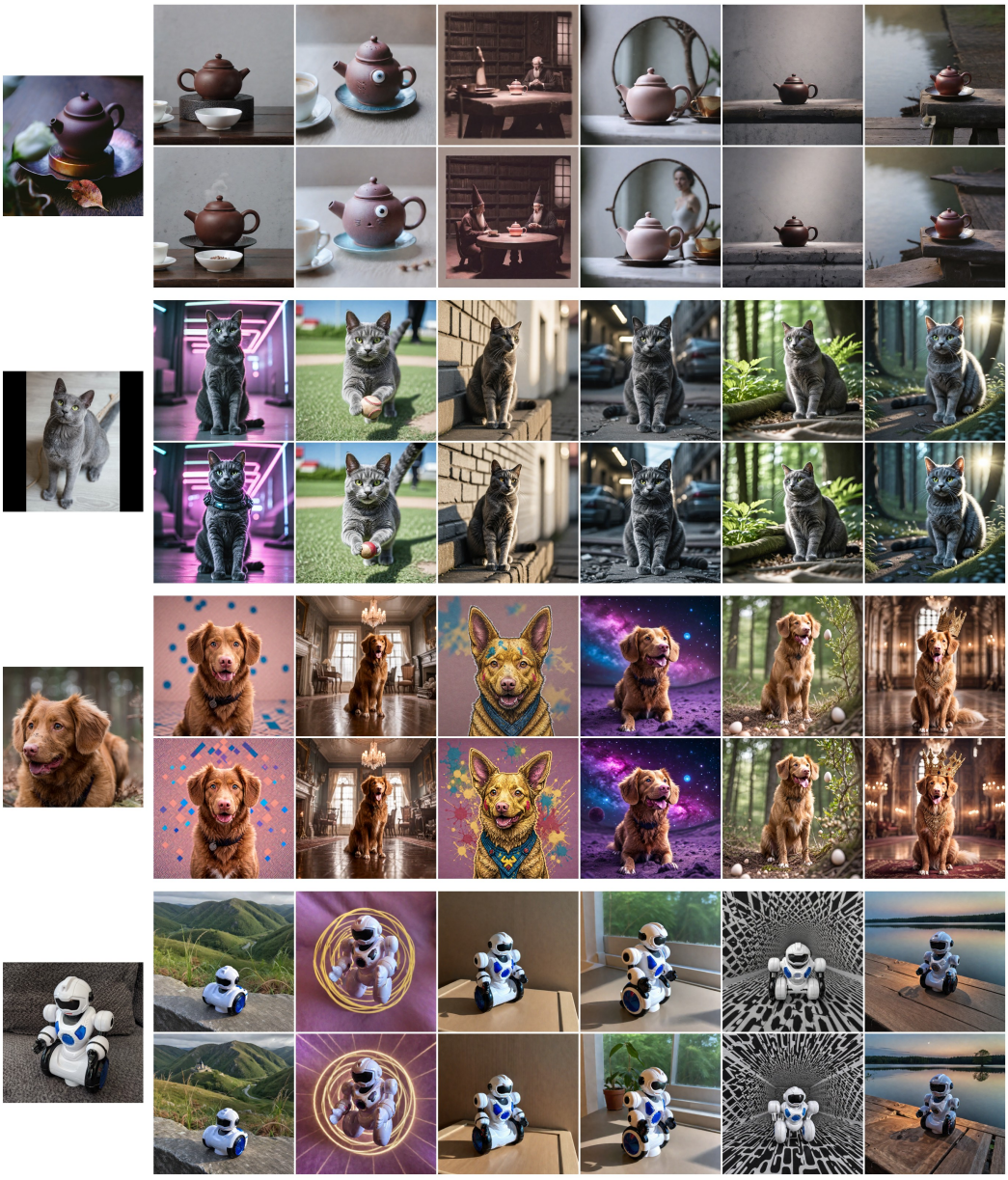

Figure 6: **DreamBooth self-improvement, full gallery (2 of 2), vanilla over guided.** Continued from Fig. 5, same layout and reference-image convention. Rows from top to bottom are teapot, cat 2, dog 7, robot toy.

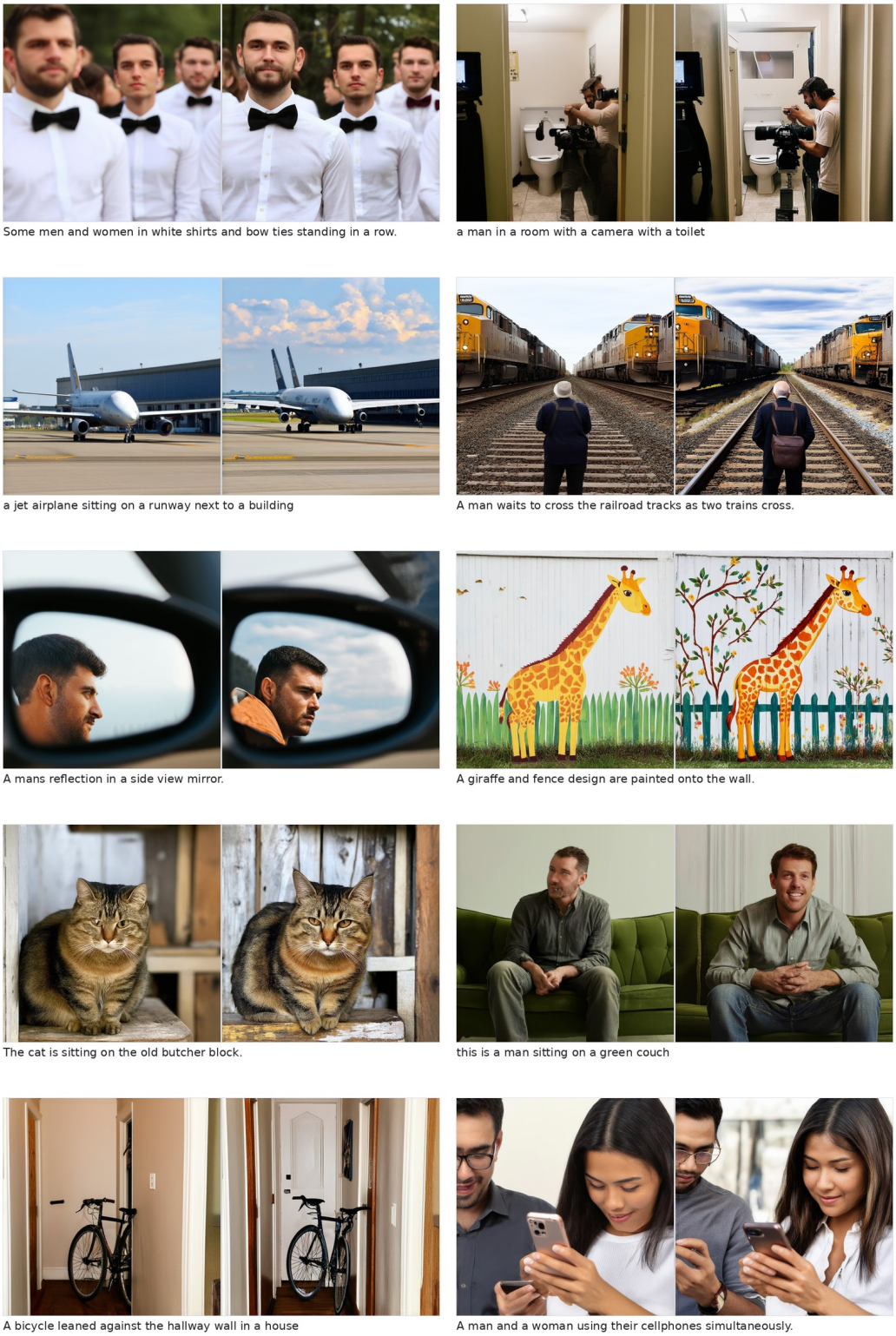

Figure 7: **Photo generation examples (SD3.5-Medium).** Ten examples; within each pair, vanilla (left) vs. our guided result (right), same prompt and seed.



Figure 8: **Anime generation examples (SD3.5-Medium).** Ten examples; within each pair, vanilla (left) vs. our guided result (right), same prompt and seed.

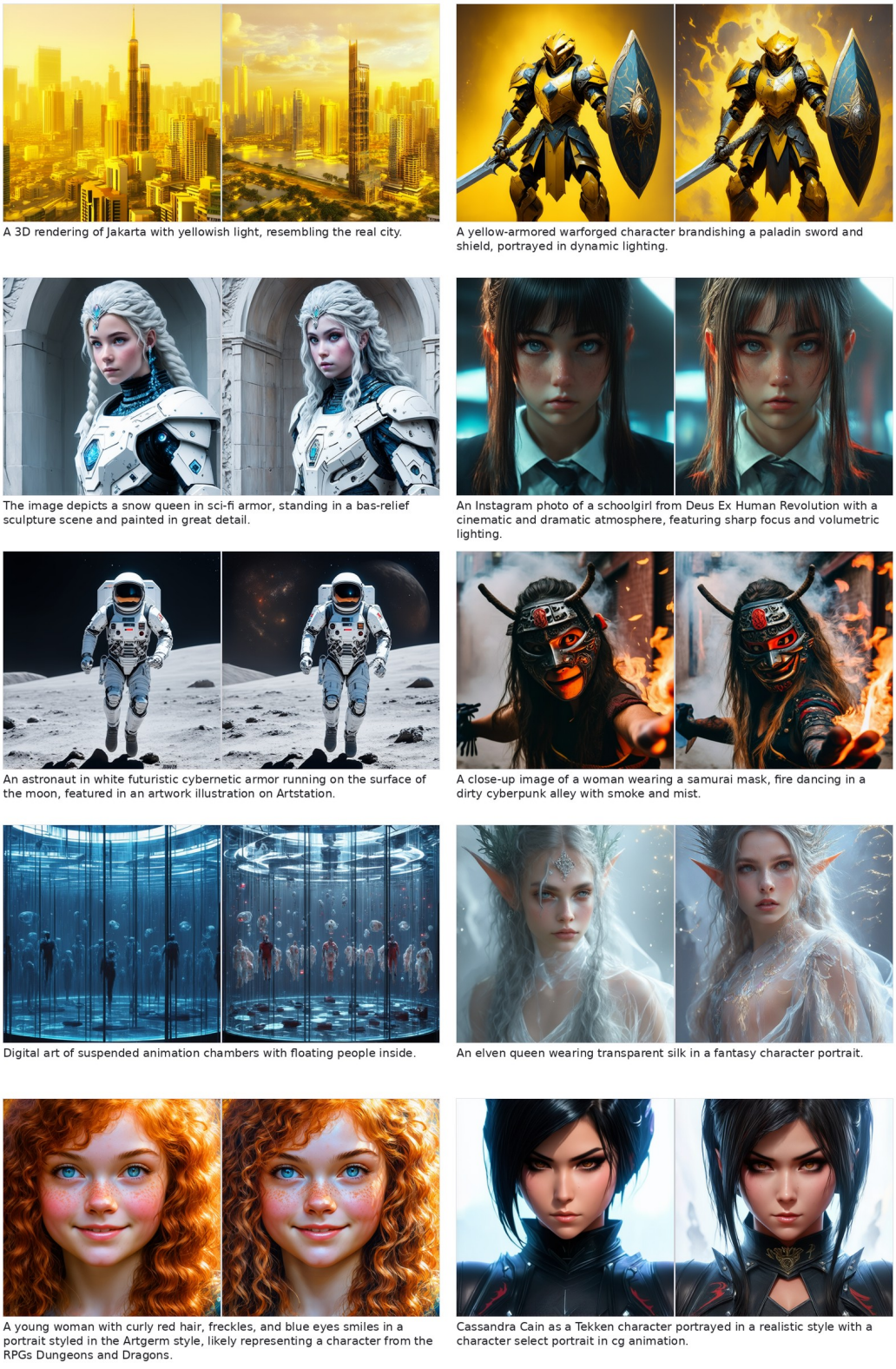

Figure 9: **Concept-art generation examples (SD3.5-Medium).** Ten examples; within each pair, vanilla (left) vs. our guided result (right), same prompt and seed.

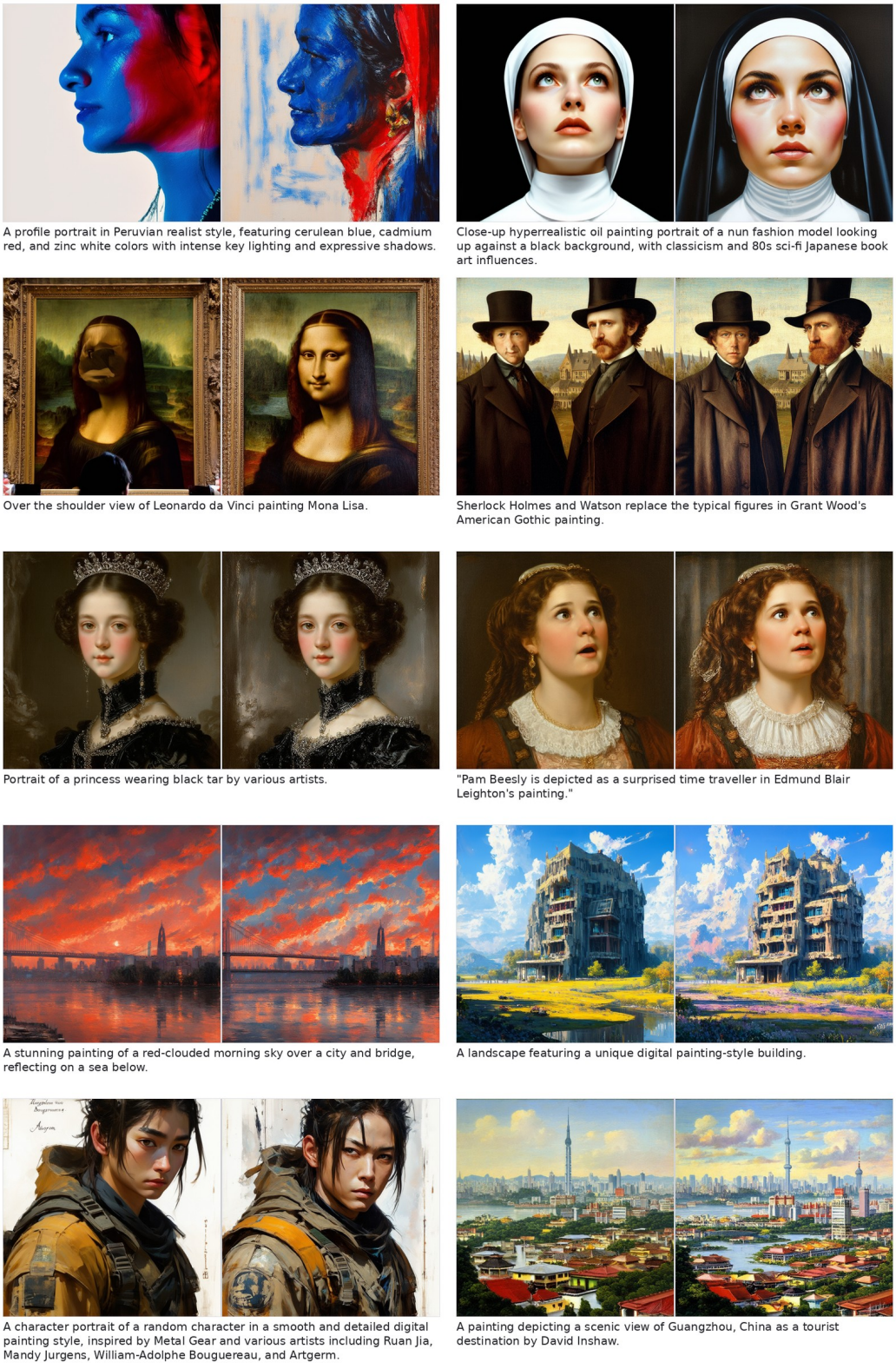

Figure 10: **Paintings generation examples (SD3.5-Medium).** Ten examples; within each pair, vanilla (left) vs. our guided result (right), same prompt and seed.

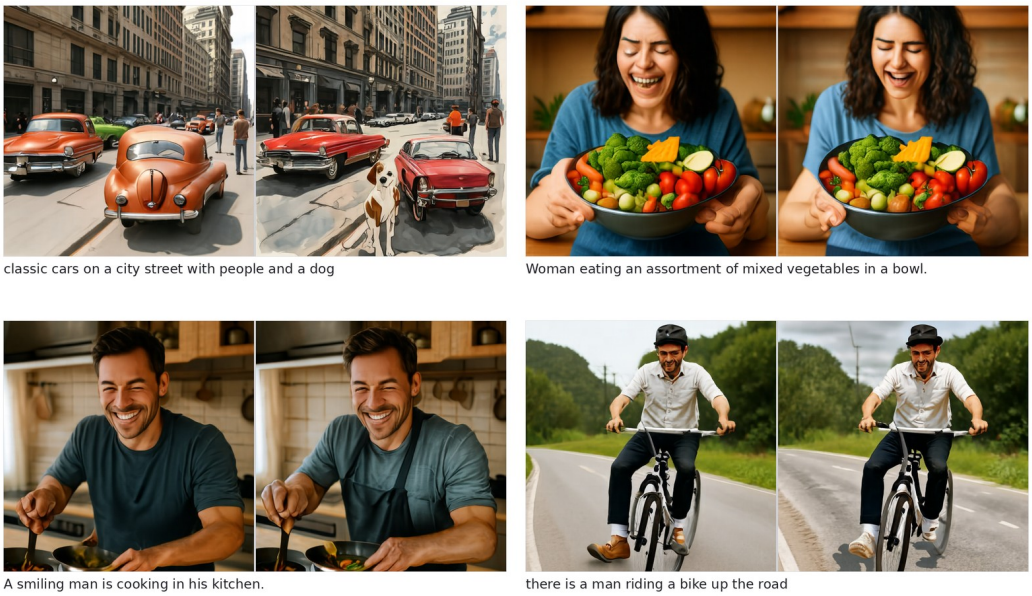

Figure 11: **Photo generation examples (PixNerd-T2I).** Within each pair, vanilla (left) vs. our guided result (right), same prompt and starting noise.

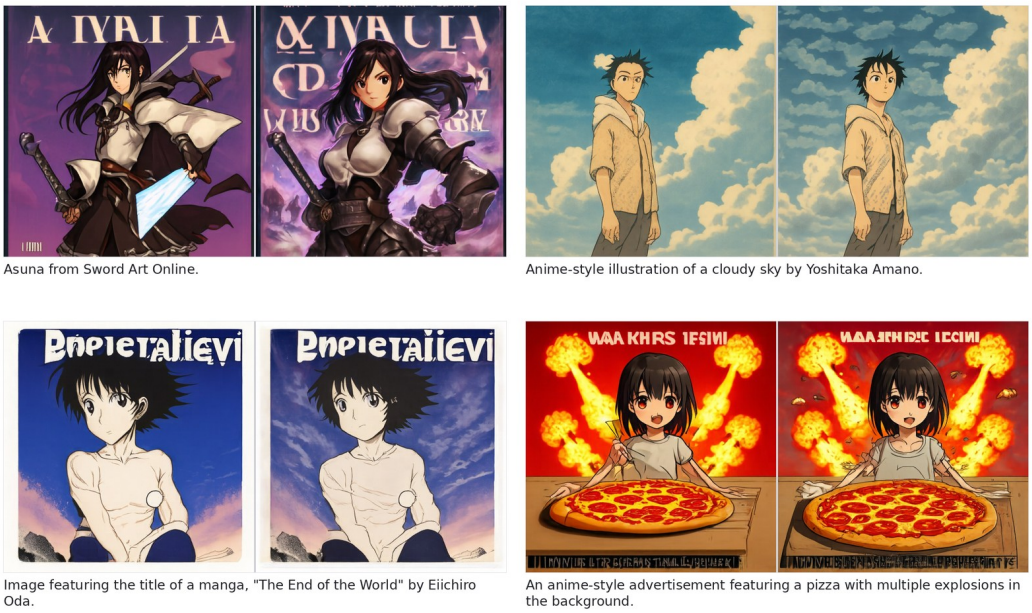

Figure 12: **Anime generation examples (PixNerd-T2I).** Within each pair, vanilla (left) vs. our guided result (right), same prompt and starting noise.

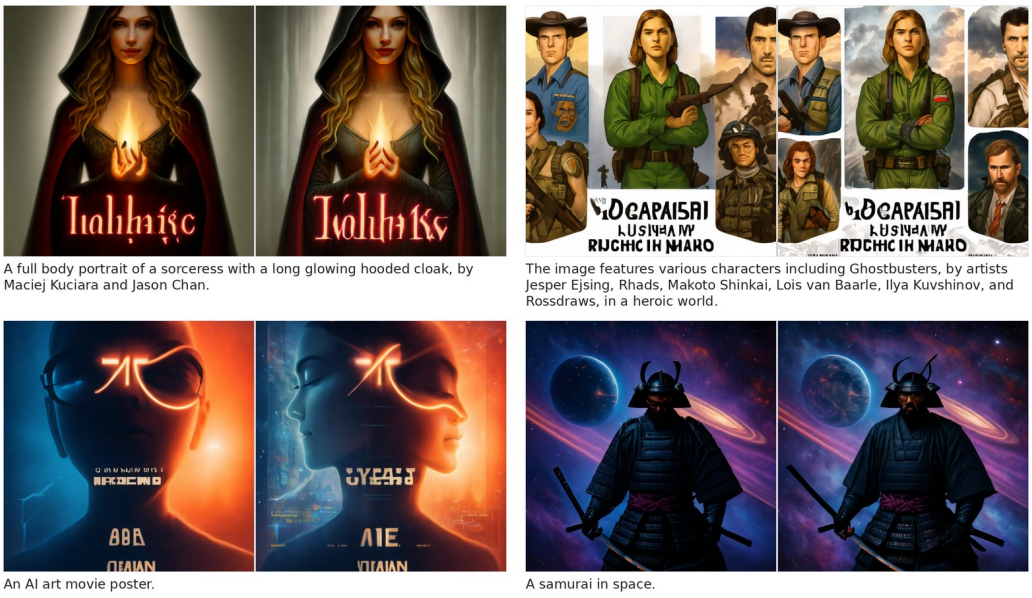

Figure 13: **Concept-art generation examples (PixNerd-T2I).** Within each pair, vanilla (left) vs. our guided result (right), same prompt and starting noise.

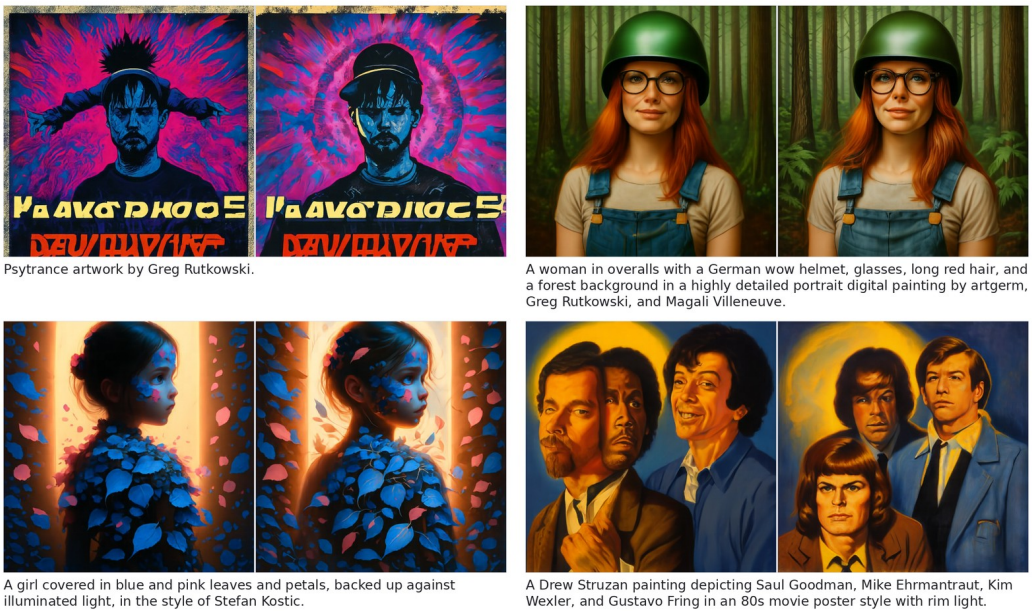

Figure 14: **Paintings generation examples (PixNerd-T2I).** Within each pair, vanilla (left) vs. our guided result (right), same prompt and starting noise.

Moment loss. Unlike the full method, the pilot matches only the first two moments of the selected directions between the generation and the fixed real-image reference (Li et al., 2015; Dziugaite et al., 2015):

$$\mathcal{L}_{\text{moment}} = \|\bar{\boldsymbol{s}} - \boldsymbol{\mu}^{\text{ref}}\|^2 + \lambda \, \|\text{cov}(\boldsymbol{s}) - \boldsymbol{\Sigma}^{\text{ref}}\|_F^2, \qquad \lambda = 0.1,$$

computed globally over all tokens of $\hat{\boldsymbol{x}}_0$, with no patch selection and no sliced transport. The gradient is taken through the clean-estimate shortcut (He et al., 2024) and normalized to the noise sphere (Yang et al., 2024). The sampler and guidance hyperparameters are listed in Table 14; no model weights are trained.

Table 14: Class-conditional ImageNet hyperparameters. The upper block is shared by all three models.

Setting	Value
feature map ϕ	DINOv3-L/16 (multi-layer sum)
selected directions m (of C)	100 (of 1024)
covariance weight λ	0.1
guidance gate σ	$[0.30, 0.95]$
Per model: sampler / strength η	
PixNerd-XL/P16	Euler 100, CFG 3.5 / η=0.003
JiT-B/16	Heun 50, CFG 3.0 / η=0.007
JiT-H/16	Heun 50, CFG 2.2 / η=0.004
generation seed (paired)	$42{,}000{,}000 + i$ (image index i)

G Human-study details

This appendix documents the two-alternative forced-choice (2AFC) preference study summarized in Table 2.

Models and stimuli. We run one study for each model type: the latent SD3.5-Medium (ImageReward-objective reference) and the pixel-space PixNerd-T2I, in each case comparing the guided output against its paired vanilla baseline (same prompt and seed). The stimuli are 40 prompts drawn at random, evenly across the four HPS v2.1 styles (ten per style), taken directly from the evaluation benchmark of Section 4.2; each pair is generated with the per-model guidance settings of Table 10.

Task and interface. Each trial shows the prompt above the two images side by side, with left/right placement randomized, and asks "Which image do you prefer?" with an explicit "cannot decide" option. The design follows Wallace et al. (2024) and Saharia et al. (2022), and trial order is shuffled per participant. The study is served by an email-gated web application (one submission per participant); recording is opt-in, and we analyze only recorded, consenting raters.

Statistics. For each model type, style, and overall we count the guided wins w, ties t ("cannot decide"), and losses l. Table 2 reports the decisive guided win rate $w/(w{+}l)$ with ties excluded (the sign-test quantity), from $n{=}13$ raters per model (520 comparisons each).

H What the method changes vs. preserves: full measurements

This appendix gives the full measurements behind Fig. 3, over all $n{=}3200$ same-prompt vanilla/guided pairs of the four-style benchmark at the SD3.5-Medium ImageReward setting of Section 4.2.

Metrics. We measure the change with quantities that capture either appearance or content. For appearance, LPIPS (Zhang et al., 2018) is a learned perceptual distance between two images (larger means they look more different). Local detail is summarized by four measures that all rise with finer texture: sharpness, the variance of the image Laplacian (a standard focus measure); total variation; gradient energy; and high-frequency spectral energy. For content, CLIP-I (Radford et al., 2021)

and DINOv2 (Oquab et al., 2024) cosine similarities compare the guided and vanilla images in a high-level feature space (1 means identical content); larger means the same scene and objects. ΔCLIP-T is the change in agreement between the image and its text prompt (Hessel et al., 2021). SSIM (Wang et al., 2004) is a structural-similarity score of at most 1 (1 means identical), comparing local luminance, contrast, and structure. Every quantity is computed per pair, between a guided image and its same-prompt, same-seed vanilla generated counterpart.

Per-metric change vs. preservation. For each measurement, Table 15 lists the vanilla value, the absolute change (abs. Δ), the relative change as a percentage of that vanilla value (rel. Δ), the paired effect size (Cohen's d_z (Cohen, 1988), the mean paired change divided by its standard deviation), and the win-rate, the fraction of the 3200 pairs whose abs. Δ has the stated sign (increase for texture and contrast, decrease for saturation). Texture rises with near-unanimous win-rates while the semantic descriptors barely move. Figure 15 plots the per-pair distributions of the two summary axes: the appearance change (LPIPS) concentrates well below the 0.84 of two unrelated images, while the content similarity (DINOv2 cosine) clusters near 1 (identical). SSIM is 0.618: sharpening and added micro-contrast lower it directly, and the high-level content measures stay close to identity (CLIP-I 0.913, DINOv2 0.827, both on a scale where 1 is the same content), so the change is in appearance rather than in the depicted scene. Prompt agreement holds for the large majority: the change ΔCLIP-T stays at or above −0.01 for 71.8% of pairs and the scene-similarity CLIP-I stays at or above 0.85 for 86.0%; the remaining pairs are the most strongly guided, where appearance moves furthest.

Table 15: What the guidance changes vs. preserves (n=3200 pairs; SD3.5-Medium at the ImageReward setting of Section 4.2). The texture and color rows report a directional change (a Wilcoxon signed-rank test on the paired differences gives $p < 10^{-15}$ for every such row except contrast); the preserved rows report a paired similarity between the vanilla and guided image, for which no direction is tested. Column definitions are in the text above.

Measurement	vanilla	abs. Δ	rel. Δ	d_z	win-rate
Changes: overall appearance					
Perceptual distance (LPIPS)	0.385	(median 0.377; 0.84 for an unrelated photo)			
Changes: texture (near-universal)					
Total variation	0.057	+0.024	+42.2%	1.85	99.2%
Gradient energy	0.259	+0.083	+32.2%	1.73	98.5%
Sharpness (Laplacian var.)	0.026	+0.016	+61.5%	0.95	97.8%
High-freq. spectral energy	0.015	+0.005	+31.4%	0.65	92.3%
Changes: color (mild)					
Saturation	0.399	−0.052	−13.0%	−0.75	75.4%
Contrast (n.s.)	0.253	+0.004	+1.6%	0.15	56.0%
Preserved: semantics (paired similarity)					
Prompt alignment ΔCLIP-T				−0.0005 (s.d. 0.021)	
Scene/object CLIP-I cos				0.913 (median 0.931)	
Semantics DINOv2 cos				0.827	
Global pixel structure (similarity, 1 = identical)					
SSIM				0.618	

The appearance-vs-content split is style-general. Table 16 repeats the measurement per style. Every style sharpens texture and moves a substantial LPIPS distance while preserving DINOv2 semantics.

I Out-of-domain style references

The reference used throughout Section 4.2 pools all four styles. A harder question is what happens when the reference comes from the wrong style entirely, for example guiding photo prompts with a reference built only from anime images. We build one single-style reference per style (SD3.5-Medium, ImageReward ranking, a pool of 3000 captions of that style) and evaluate all 16 reference×target pairings under the settings of Section 4.2 (η=0.008, ρ=0.25, deterministic paired sampling, 800 prompts per target style), prepared together so the cells are directly comparable.

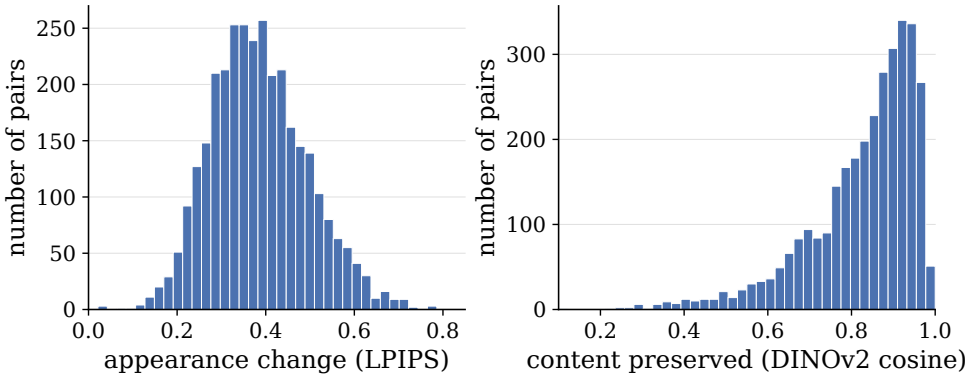

Figure 15: **Appearance changes while content is preserved** (all n=3200 vanilla/guided pairs at the +0.072 ImageReward setting). **Left:** the appearance change (LPIPS) is concentrated around 0.38, well below the 0.84 of two unrelated photos. **Right:** content similarity (DINOv2 cosine) is concentrated near 1.0 (identical), with 84% of pairs above 0.7.

Table 16: The appearance-vs-content split is style-general (SD3.5-Medium at the ImageReward setting of Section 4.2, 800 paired prompts per style). Sharpness is reported as the vanilla value, the absolute change, and the relative change, alongside the appearance distance (LPIPS) and content similarity (DINOv2 cosine).

Style	Sharpness (van.)	abs. Δ	rel. Δ	LPIPS	DINOv2 cos
Photo	0.023	+0.016	+68.3%	0.377	0.817
Anime	0.029	+0.013	+43.6%	0.393	0.842
Concept-art	0.023	+0.016	+69.4%	0.386	0.826
Paintings	0.028	+0.019	+68.1%	0.386	0.821
All	0.026	+0.016	+61.5%	0.385	0.827

Table 17 shows that every pairing improves its target. The cost of a mismatched reference falls mostly on the photo target, where the in-domain reference beats the out-of-domain average by 0.017 ImageReward. On the three artistic targets the in-domain and out-of-domain averages differ by at most 0.009, and the photo reference is best or within 0.004 of the best in every column. One possible reason is that photo is the only style that targets photographic realism, so its reference is the only one whose feature statistics resemble natural images, while the three artistic styles share enough of their statistics that the reference's style hardly matters. In practice this lowers the bar for applying the method, since the reference pool does not have to match the target domain.

Table 17: Out-of-domain style references, ΔImageReward versus the target style's baseline (SD3.5-Medium, ImageReward objective, 800 prompts per target). Rows are the style the reference is built from, columns the style being generated; diagonal (in-domain) cells are bolded.

Reference ↓ Target →	Photo	Anime	Concept-art	Paintings
Photo	**+0.090**	+0.050	+0.080	+0.060
Anime	+0.076	**+0.053**	+0.065	+0.057
Concept-art	+0.069	+0.038	**+0.058**	+0.064
Paintings	+0.075	+0.044	+0.053	**+0.058**

J FAILURE MODE: OVER-GUIDANCE

Guidance with a single global strength η can push too hard on prompts that needed no correction; we call this over-guidance. Although the mean changes in Table 3 are non-negative, in every style only about 60% of individual prompts improve. This appendix examines the regressing minority,

focusing on the ImageReward objective on n=3200 same-prompt vanilla/guided SD3.5-Medium pairs (guidance settings of Table 10).

Over-guidance moves the image away from its prompt. We compare the 50 most negative pairs per style by ΔImageReward (the failure group, totally 200 pairs across 4 styles) against the 50 most positive (the success group, 200 pairs). A vision-language model (Qwen3-VL-32B-Instruct) sees each pair side by side, with the guided image placed left or right at random, and answers which half matches the prompt better (LEFT, RIGHT, or SAME; temperature 0). The judge is never told which half is the guided one. The two groups separate sharply (Table 18). The guided image is judged the worse match in 50% of failure pairs and 24% of success pairs ($p < 10^{-7}$, two-sided Fisher's exact test), with the same direction in every style and within both left/right placements. The judge reads the prompt just as ImageReward does, so its agreement is convergent evidence from an architecturally unrelated model, not a fully independent measure.

Table 18: How often the guided image is judged the worse match for its prompt. The question reads: "Which picture matches the prompt better, considering the objects it asks for and their counts?" SAME answers (about 6%) count toward the denominator.

Guided judged worse match	photo	anime	concept-art	paintings	total
Failure group (50 per style)	46%	50%	56%	48%	50%
Success group (50 per style)	18%	30%	22%	24%	24%

Figure 16 shows eight failure pairs. The changes are easy to overlook. An object the prompt asks for disappears or changes count, the scene is recomposed, a face or a patch of text degrades, or the tone shifts. For instance, a prompt asking for "two vespas" returns one, a pattern asked to tile "seamlessly" acquires seams, and "fruit goblins with expressive eyes" lose their faces.

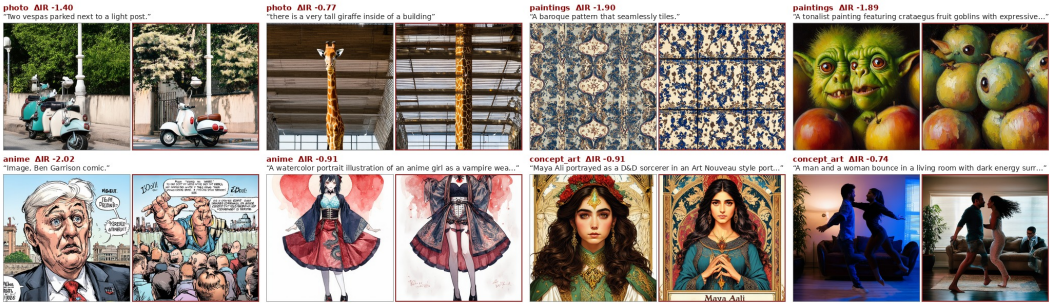

Figure 16: **Failure mode of over-guidance (vanilla | guided).** Two of each style's worst failure pairs by ΔImageReward; vanilla left (grey border), guided right (red). Same prompt and starting noise within each pair. The guided images are rarely unrecognizable; they depict a slightly different scene.

A concrete habit: inserted clouds. A recurring content change in the photo style is extra cloud. We asked the same judge which half of each photo pair shows more clouds in its sky. Among the 254 pairs with visible sky, it names the guided image in 76% and the vanilla image in 16%, as often in successful pairs as in failing ones. The extra cloud is a general signature of the guidance in this style, not what separates the failures. Figure 17 shows four such pairs. A plausible reason is that the target metric itself favors textured, dramatic skies over flat ones, so the guidance pushes sky regions toward cloud.

These content changes are too small for image-level metrics to see. In the failing minority the edit produces unwanted content but stays small, and what remains is overwhelmingly the same scene. Specifically, between the failure and success groups the median perceptual distance (LPIPS) is 0.43 against 0.45, and the median semantic similarity (DINOv2 cosine) is 0.78 against 0.79; neither separates the groups.



Figure 17: **Inserted clouds (vanilla | guided).** Four photo pairs where the judge names the guided image as the cloudier one; vanilla left (grey border), guided right (orange). Same prompt and starting noise within each pair.

Thus, the difference is not how much the image changes but what changes. When the affected object is one the prompt requires, only a prompt-reading metric notices. The image-to-image rows of Table 15 (CLIP-I, DINOv2, SSIM) therefore certify that the depicted scene survives guidance but not that each prompt is still satisfied.

The guidance inherits the metric's taste, not its prompt-awareness. This looks paradoxical. We optimize for ImageReward, a prompt-aware metric, yet the failures move away from the prompt. This is due to where the metric enters in our method. ImageReward is used once, offline, to rank the reference pool (Section 3.3). The guidance term itself never reads the prompt and only pushes patch statistics toward the appearance shared by highly ranked images. That distillation keeps what high-reward images have in common, their look, and discards what made each one high-reward for its own prompt. The guidance edits the patches whose local statistics sit farthest from the reference, without checking what those patches contain. When they contain an object the prompt requires, the guidance still repaints them toward the reference look, and the repainted object may no longer be what the prompt asked for. The inserted clouds of Fig. 17 fit the same picture. If textured skies are common in high-reward photos, the guidance will add them whether or not a prompt asks.

Guidance hurts the images that were already sharp. The failure and success groups of Table 18 also differ before guidance is applied. Measured on the vanilla images as the variance of the image Laplacian (Section H), the failure group's median sharpness is 0.022, against 0.016 for the success group and 0.018 over all pairs (Table 15 reports the mean, 0.026). Thus, guidance helps when it starts from a soft image and adds detail; it hurts when the detail is already there and the same step is applied anyway. The vanilla halves of Fig. 16 illustrate this, most of them already rich in detail. We conclude that the single global η is not too strong in general but too strong for those prompts. We leave adapting the strength per image to future work.

K Failure mode: the projector tax

Section 4.2 shows the same guidance helping the pixel-space PixNerd-T2I less than the latent models, and Section 5 mentions that even a latent model loses almost all of its gain when guided through a foreign encoder. This section offers a common explanation for both. They are two payments of the same price, made by any model when its guidance must read images through an autoencoder that is not its own.

The guidance sees PixNerd's images through a borrowed lens. PixNerd-T2I generates pixels directly and has no autoencoder, so our guidance borrows SD3.5-Medium's, encoding each clean estimate into the borrowed latent space before any feature can be read and returning the correction as the difference of the two decoded images (Section B). Whatever the borrowed autoencoder cannot represent is invisible to that decision.

How much the borrowed lens loses. We measure the loss with a round trip: encode an image with the borrowed autoencoder, decode it straight back, and compare the result with the input. On 400 PixNerd images the round trip returns 34.6 dB PSNR and loses 16.8% of the images' sharpness; on 400 of SD3.5-Medium's own images, which this autoencoder was built to represent, it returns 37.1 dB and loses only 7.3%. Thus, borrowing costs 2.5 dB, and relative to its own images the

borrowed autoencoder destroys about 2.3× more of PixNerd's fine detail. The loss is too fine to see at full-image scale, and that is exactly the problem; it sits in the same texture the guidance is trying to improve, so it never reaches the guidance and cannot be corrected by it, consistent with the smaller pixel-space gains in Table 3. The guidance still helps PixNerd on average, and Figs. 11 to 14 show examples of its stronger gains.

The pixel-space model fails more. For the ImageReward objective, PixNerd-T2I regresses more often than SD3.5-Medium, on 46% of prompts against 40%, and its failures cut deeper into prompt alignment. Figure 18 shows two of each style's worst pairs.

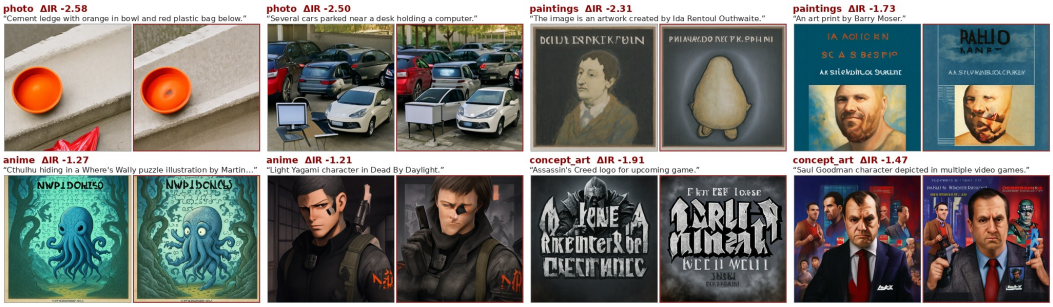

Figure 18: **PixNerd-T2I failure examples (vanilla | guided).** Two of each style's worst failure pairs by ΔImageReward; vanilla left (grey border), guided right (red). Same prompt and starting noise within each pair.

The prompt-match judge of Section J agrees (Table 19). On PixNerd's failure group, formed as in Section J, the guided image is judged the worse match in 55% of pairs, close to the latent model's 50%; on the success group it is still judged worse in 35% of pairs against the latent model's 24%, so even PixNerd's winning edits often turn out to be worse matches.

Table 19: The same prompt-match judgment as Table 18, on PixNerd-T2I; how often the guided image is judged the worse match. Groups are the 50 most negative and most positive pairs per style by ΔImageReward, judged as in Table 18; SAME answers (about 7%) count toward the denominator.

Guided judged worse match	photo	anime	concept-art	paintings	total
Failure group (50 per style)	58%	62%	52%	46%	55%
Success group (50 per style)	34%	38%	38%	30%	35%

Why the pixel model fails more is not fully clear to us. Our guess is about where each correction is applied. SD3.5-Medium's correction is applied inside its latent space, and everything in that space decodes to a typical SD3.5-Medium image, so even a badly aimed step still yields a normal-looking picture. PixNerd's correction is added directly to pixel values, and an arbitrary pixel edit is not guaranteed to look like a photograph, so a badly aimed step can leave visible damage. A full account, and pixel-space guidance that closes this gap, is left to future work.