REWARD TILTING VIA SEMANTIC NOISING

Anonymous authors
Paper under double-blind review

ABSTRACT

Steering a generative model toward a desired property typically requires separate fine-tuning for each reward or additional computation during sampling. We propose *semantic noising*, which conditions generation on a noisy semantic representation of the data. The resulting posteriors correspond to exponential tilts by linear-quadratic rewards in semantic space, allowing a single model to support objectives and strengths chosen after training. A conditional generator can be trained to approximate these posteriors directly, while models that generate semantic representations by diffusion provide analytic tilted denoisers for the same family of rewards. We illustrate semantic noising in three settings. First, we train MolGPT, an autoregressive molecular generator, to condition on noisy semantic representations without using molecular property values, and use independently fitted linear predictors to control properties. Our model achieves a better reward–KL trade-off than separately trained property-conditioned models. Second, our joint diffusion model of images and their CLIP embeddings achieves FID comparable to REG on ImageNet while enabling prompt-based editing and adjustable aesthetic-score tilting. Finally, we show that pretrained factorized models such as Kandinsky support analytic tilting without retraining or additional denoiser evaluations.

1 INTRODUCTION

In many applications, we want a generative model to favor particular properties, such as greater aesthetic quality in images or a desired molecular attribute. The objective and its desired strength are often chosen only after the model has been trained. Existing approaches commonly fine-tune the model separately for each reward (Lu et al., 2023; Liu et al., 2025; Domingo-Enrich et al., 2025; Bergmeister et al., 2026; Potaptchik et al., 2026) or add computation during sampling (Skreta et al., 2025; Singhal et al., 2025), making it costly to accommodate different objectives. We aim to support a useful family of rewards with a single model, without repeating this work for each reward.

For a given reward function $r(\mathbf{x})$, we seek a distribution that favors higher-reward examples while staying close to the base distribution $q(\mathbf{x})$. Maximizing expected reward penalized by KL divergence from q is the exponential tilt (Donsker & Varadhan, 1983),

$$q^r(\mathbf{x}) \propto q(\mathbf{x}) \exp r(\mathbf{x}).$$

Efficient sampling from q^r is generally difficult for arbitrary r, and we would rather not solve a new sampling problem for every reward.

Many useful objectives take a simple form in a suitable representation of the data, i.e., the *semantic space*. For example, a reward that favors proximity in this space prefers images with similar content even when their pixels differ, while a linear projection can predict properties such as aesthetic quality or molecular attributes. We show how to efficiently tilt by such rewards via *semantic noising*, conditioning generation on a noisy semantic representation. By Bayes' rule, conditioning on the noisy representation reweights q by a Gaussian likelihood, which is the exponential of a reward that is linear-quadratic in semantic space. The choice of representation therefore defines the reward family, while the noisy representation and its noise level specify a particular reward within it.

First, we show that the posteriors induced by semantic noising can be learned directly with a conditional generative model. For each training example, we compute its semantic representation and add Gaussian noise. The noisy representation and its noise level serve as inputs, with the original example as the training target. This approach applies to any conditional generative model and does

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.

not require choosing the downstream reward in advance. At inference time, we choose the noisy representation input and the noise level to specify the desired reward. A property predictor used to define a reward can be fitted independently without retraining the generator.

Analytic tilting with semantic diffusion. Second, diffusion models that generate semantic representations offer another way to sample from the same family of tilted distributions. For linear-quadratic rewards, the tilted denoiser can be expressed analytically in terms of the original denoiser at no extra cost (Liu, 2024; Holderrieth et al., 2026; Moitra et al., 2026). Existing models that jointly diffuse data and semantic representations have been shown to improve generation quality and training efficiency (Bao et al., 2023; Kouzelis et al., 2025; 2026; Wu et al., 2025). We show that a small change to their training, sampling the data and semantic noise levels independently, enables these models to support the proposed family of rewards. A second option uses models such as unCLIP and Kandinsky (Ramesh et al., 2022; Razzhigaev et al., 2023), where a diffusion prior generates a semantic representation and a separate decoder generates the data conditioned on that representation. Such models support these rewards without retraining either component.

We evaluate both approaches in molecular and image generation. We train MolGPT, an autoregressive molecular generator (Bagal et al., 2021), to condition on noisy CLAMP embeddings (Seidl et al., 2023) without using any property values. Independently fitted linear predictors provide reward directions that allow the same generator to control properties individually and jointly. The proposed model reaches the same mean property values at lower KL divergence from the base distribution than separately trained property-conditioned models. For diffusion-based image generation, our joint model achieves FID comparable to REG (Wu et al., 2025) on ImageNet (Deng et al., 2009) while enabling prompt-based editing and adjustable aesthetic-score tilting, with negligible computational overhead. Finally, a pretrained Kandinsky model supports semantic tilting without retraining or additional denoiser evaluations.

2 Reward tilting via semantic noising

Let $q(\mathbf{x})$ be the data distribution. We seek to maximize a reward $r(\mathbf{x})$ while remaining close to q, as measured by the KL divergence, which is uniquely solved by the *exponential tilt*

$$q^r := \arg\max_p \{\mathbb{E}_p[r(\mathbf{x})] - D_{\mathrm{KL}}[p\|q]\} \quad \Longrightarrow \quad q^r(\mathbf{x}) \propto q(\mathbf{x}) \exp r(\mathbf{x}). \tag{1}$$

We consider rewards defined on a semantic representation $\mathbf{z} = \boldsymbol{\phi}(\mathbf{x}) \in \mathbb{R}^k$, where $\boldsymbol{\phi}$ is a fixed feature map. The features may come from a pretrained encoder, such as CLIP (Radford et al., 2021) for images, or be hand-designed. Our construction relates a family of rewards in semantic space to conditioning on Gaussian-corrupted representations.

2.1 Posteriors induced by semantic noising

We define *semantic noising* by adding Gaussian noise to the semantic representation. For a noise precision $\tau_{\mathbf{z}} > 0$, we use the noisy representation $\boldsymbol{\phi}(\mathbf{x}) + \boldsymbol{\varepsilon}/\sqrt{\tau_{\mathbf{z}}}$ scaled by $\tau_{\mathbf{z}}$

$$\tilde{\mathbf{z}} = \tau_{\mathbf{z}} \boldsymbol{\phi}(\mathbf{x}) + \sqrt{\tau_{\mathbf{z}}}\, \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}). \tag{2}$$

Expanding the Gaussian likelihood and applying Bayes' rule identifies the posterior as an exponential tilt with a linear-quadratic semantic reward. For any $\mathbf{a} \in \mathbb{R}^k$ and $\gamma > 0$, we have

Semantic noising as reward tilting

$$q(\mathbf{x} \mid \tilde{\mathbf{z}} = \mathbf{a}, \tau_{\mathbf{z}} = \gamma) \propto q(\mathbf{x}) \exp\Big(\mathbf{a} \cdot \boldsymbol{\phi}(\mathbf{x}) - \tfrac{\gamma}{2}\|\boldsymbol{\phi}(\mathbf{x})\|^2\Big) = q(\mathbf{x}) \exp(r_{\mathbf{a},\gamma}(\mathbf{x}))\,. \tag{3}$$

The choice of feature map $\boldsymbol{\phi}$ determines which rewards can be expressed, while the conditioning parameters $(\tilde{\mathbf{z}}, \tau_{\mathbf{z}})$ specify a member of this family. At $(\tilde{\mathbf{z}}, \tau_{\mathbf{z}}) = (\mathbf{0}, 0)$, the posterior reduces to q.

If $\boldsymbol{\phi}(\mathbf{x})$ has constant norm, the quadratic term is constant across examples and does not affect the tilted distribution. The family then reduces to rewards that are linear in $\boldsymbol{\phi}(\mathbf{x})$, and the posterior no longer depends on $\tau_{\mathbf{z}}$ once $\tilde{\mathbf{z}}$ is fixed, i.e. $q(\mathbf{x} \mid \tilde{\mathbf{z}}, \tau_{\mathbf{z}}) = q(\mathbf{x} \mid \tilde{\mathbf{z}})$.

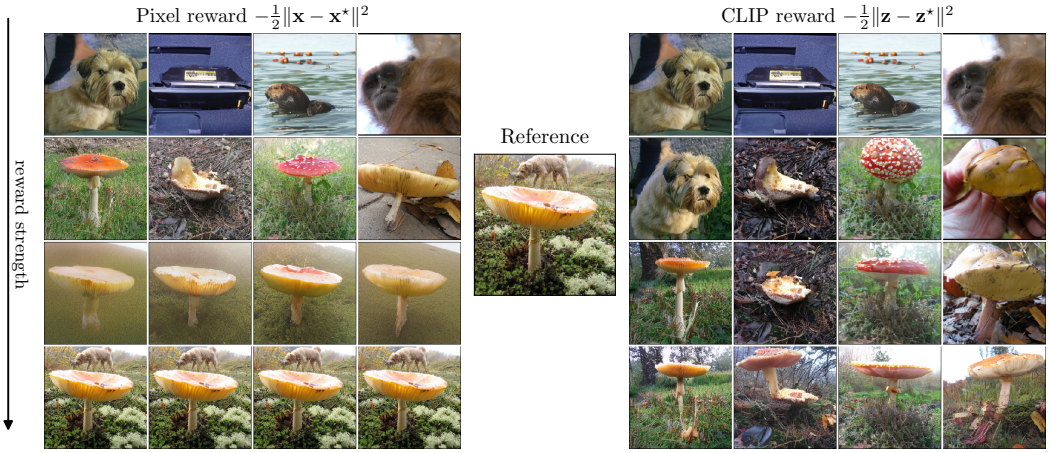

Figure 1: **Tilting toward a reference in semantic space.** Samples from our joint diffusion model (Section 4.2) tilted toward a reference image (center) using the pixel-space reward $-\frac{\lambda}{2}\|\mathbf{x}-\mathbf{x}^\star\|^2$ (left) or the CLIP-space reward $-\frac{\lambda}{2}\|\mathbf{z}-\mathbf{z}^\star\|^2$ (right), with λ increasing from top to bottom. The pixel-space reward pulls samples toward the reference image itself, whereas the CLIP-space reward favors diverse images with similar semantic content.

2.2 Specifying Rewards

A reference point $\mathbf{z}^\star$ in semantic space and a reward strength λ define the attraction reward

$$r(\mathbf{x}) = -\tfrac{\lambda}{2}\|\boldsymbol{\phi}(\mathbf{x}) - \mathbf{z}^\star\|^2, \tag{4}$$

which is equivalent to $r_{\mathbf{a},\gamma}$, with linear coefficient $\mathbf{a} = \lambda\mathbf{z}^\star$ and quadratic coefficient $\gamma = \lambda$. It favors examples whose semantic representations are close to the reference. The reference may come from another data example or, in a shared image-text semantic space, from a text prompt. Figure 1 contrasts the proximity reward in pixel space and CLIP space.

Linear property predictors $r(\mathbf{x}) = \mathbf{a}\cdot\boldsymbol{\phi}(\mathbf{x})$ are another useful class of rewards. Weighted sums of such rewards are specified by the corresponding weighted sums of their directions in semantic space, allowing simultaneous control of multiple properties without retraining the generator or fitting new predictors for each combination.

2.3 Direct Conditional Generation

The posterior identity in Eq. 3 suggests a direct training procedure. For each example $\mathbf{x}\sim q$, construct $\tilde{\mathbf{z}}$ from Eq. 2, and train a conditional generative model with its usual objective,

$$p_{\boldsymbol{\theta}}(\mathbf{x} \mid \tilde{\mathbf{z}}, \tau_{\mathbf{z}}) \approx q(\mathbf{x} \mid \tilde{\mathbf{z}}, \tau_{\mathbf{z}}). \tag{5}$$

Training does not require choosing a downstream reward in advance. At inference time, for $r_{\mathbf{a},\gamma}$ with $\gamma > 0$, we set $\tilde{\mathbf{z}} = \mathbf{a}$ and $\tau_{\mathbf{z}} = \gamma$, so that $p_{\boldsymbol{\theta}}(\mathbf{x} \mid \tilde{\mathbf{z}} = \mathbf{a}, \tau_{\mathbf{z}} = \gamma) \approx q(\mathbf{x})\exp(r_{\mathbf{a},\gamma}(\mathbf{x}))$. This procedure applies to any conditional generative model. We demonstrate it with an autoregressive model applied to molecular generation in Section 4.1.

When $\|\boldsymbol{\phi}(\mathbf{x})\| = const$, the model can instead learn $p_{\boldsymbol{\theta}}(\mathbf{x} \mid \tilde{\mathbf{z}})$ and omit $\tau_{\mathbf{z}}$ from the conditioning input, which does not affect the target distribution.

3 Analytic Tilting with Semantic Diffusion

Diffusion models that generate semantic representations provide another way to sample the tilted distributions of Section 2. For these models, the tilted denoiser is obtained analytically by shifting the inputs of the original denoiser. We first review this identity, then apply it to joint diffusion of data and semantic representations and to a semantic diffusion prior followed by a conditional generator.

3.1 Primer on diffusion models and natural coordinates

A Gaussian diffusion model for a data distribution $q(\mathbf{x})$ defines noisy variables $\mathbf{x}_t = \alpha_t \mathbf{x} + \sigma_t \boldsymbol{\varepsilon}$ for $t \in [0, T]$, where the signal-to-noise ratio $\mathrm{SNR}(t) = \alpha_t^2/\sigma_t^2$ decreases with t (Song et al., 2021; Ho et al., 2020; Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023; Liu et al., 2023). The corresponding marginals are

$$p_t(\mathbf{x}_t) = \int q(\mathbf{x}) \mathcal{N}(\mathbf{x}_t \mid \alpha_t \mathbf{x}, \sigma_t^2 \boldsymbol{I}) d\mathbf{x}.$$

For $f_t = \partial_t \log \alpha_t$ and $g_t^2 = -\sigma_t^2 \partial_t \log \mathrm{SNR}(t)$, the denoising processes are

$$\begin{aligned} \text{Reverse SDE:} \quad & d\mathbf{x}_t = \big(f_t \mathbf{x}_t - g_t^2 \nabla \log p_t(\mathbf{x}_t)\big)\, dt + g_t d\overline{W}_t, \\ \text{PF ODE:} \quad & d\mathbf{x}_t = \big(f_t \mathbf{x}_t - \tfrac{1}{2} g_t^2 \nabla \log p_t(\mathbf{x}_t)\big)\, dt, \end{aligned}$$

where $\overline{W}$ is the Brownian motion running backwards in time, and $\nabla \log p_t$ is the score function approximated by a model (Hyvärinen, 2005; Vincent, 2011). Initializing $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \sigma_T^2 \boldsymbol{I})$ and solving either process yields an approximate sample at $t = 0$. The probability-flow ODE is deterministic and can also be integrated forward to encode a data point, which we use in our editing experiments.

By Tweedie's formula (Efron, 2011), the score satisfies $\sigma_t^2 \nabla \log p_t(\mathbf{x}_t) = \alpha_t \mathbb{E}[\mathbf{x} \mid \mathbf{x}_t, t] - \mathbf{x}_t$. Both processes therefore depend on q only through the denoiser $D(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{x} \mid \mathbf{x}_t, t]$.

Karczewski et al. (2026) observed that the denoising posteriors form an exponential family with natural coordinates

$$\tilde{\mathbf{x}} = \frac{\alpha_t \mathbf{x}_t}{\sigma_t^2}, \quad \tau = \frac{\alpha_t^2}{\sigma_t^2}. \tag{6}$$

In these coordinates, we have $\tilde{\mathbf{x}} = \tau \mathbf{x} + \sqrt{\tau}\, \boldsymbol{\varepsilon}$, the same parameterization as semantic noising in Section 2. Unlike the standard coordinates $(\mathbf{x}_t, t)$, the natural coordinates do not depend on the choice of noise schedule, and they coincide with the coordinates of the associated stochastic localization process (Montanari, 2023). We write $D(\tilde{\mathbf{x}}, \tau) \coloneqq \mathbb{E}[\mathbf{x} \mid \tilde{\mathbf{x}}, \tau]$ for the denoiser in these coordinates.

In practice, natural coordinates are numerically unstable near $t = 0$, so we parametrize models in standard coordinates. These are recovered from $(\tilde{\mathbf{x}}, \tau)$ by $t = \mathrm{SNR}^{-1}(\tau)$ and $\mathbf{x}_t = \alpha_t \tilde{\mathbf{x}}/\tau$. We provide further background in Appendix A and Appendix B.

3.2 Linear-quadratic tilting

For a tilted data distribution q^r, the corresponding denoiser is given by

$$D^r(\tilde{\mathbf{x}}, \tau) = \frac{\mathbb{E}\big[\mathbf{x} \exp r(\mathbf{x}) \mid \tilde{\mathbf{x}}, \tau\big]}{\mathbb{E}\big[\exp r(\mathbf{x}) \mid \tilde{\mathbf{x}}, \tau\big]},$$

which is generally intractable. For a linear-quadratic reward, however, multiplying the denoising posterior by $\exp r(\mathbf{x})$ shifts its natural parameters, so the tilted denoiser satisfies (McDonald, 2026)

$$r(\mathbf{x}) = \mathbf{a} \cdot \mathbf{x} - \tfrac{\gamma}{2} \|\mathbf{x}\|^2 \quad \Longrightarrow \quad D^r(\tilde{\mathbf{x}}, \tau) = D(\tilde{\mathbf{x}} + \mathbf{a}, \tau + \gamma), \tag{7}$$

for $\gamma \geq 0$. This includes the purely linear case $\gamma = 0$ (Moitra et al., 2026) and the quadratic proximity reward $-\frac{\lambda}{2}\|\mathbf{x} - \mathbf{x}^\star\|^2$, which corresponds to $\mathbf{a} = \lambda \mathbf{x}^\star$ and $\gamma = \lambda$ (Liu, 2024). We provide a proof in natural coordinates and discuss the connection to posterior sampling in Appendix B.1.

Substituting this denoiser into the reverse SDE or probability-flow ODE gives a sampling process for the tilted distribution. The shift requires neither reward gradients nor additional denoiser evaluations. The identity holds for the exact conditional mean, while in practice the shifted evaluations use a learned approximation to it.

3.3 Joint diffusion of data and semantic representations

Several approaches jointly diffuse data and their semantic representations to improve generation quality or speed up convergence (Bao et al., 2023; Kouzelis et al., 2025; 2026; Wu et al., 2025). We adapt this setup by sampling the data and semantic noise levels independently during training.

Let $q(\mathbf{x}, \mathbf{z})$ be the joint distribution with $\mathbf{z} = \phi(\mathbf{x})$. We apply Gaussian corruption independently to the two coordinates,

$$\mathbf{x}_{t_\mathbf{x}} = \alpha_{t_\mathbf{x}}\mathbf{x} + \sigma_{t_\mathbf{x}}\boldsymbol{\varepsilon}_\mathbf{x}, \quad \mathbf{z}_{t_\mathbf{z}} = \alpha_{t_\mathbf{z}}\mathbf{z} + \sigma_{t_\mathbf{z}}\boldsymbol{\varepsilon}_\mathbf{z},$$

with independent diffusion times $t_\mathbf{x}, t_\mathbf{z}$ and independent noise $\boldsymbol{\varepsilon}_\mathbf{x}, \boldsymbol{\varepsilon}_\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. Each coordinate has its own natural parameters $(\tilde{\mathbf{x}}, \tau_\mathbf{x})$ and $(\tilde{\mathbf{z}}, \tau_\mathbf{z})$, giving a joint denoiser $(D_\mathbf{x}, D_\mathbf{z})(\tilde{\mathbf{x}}, \tau_\mathbf{x}, \tilde{\mathbf{z}}, \tau_\mathbf{z})$. For the semantic reward $r_{\mathbf{a},\gamma}$ (Eq. 3), the tilted joint denoiser shifts only the semantic inputs (proof in Appendix B.2),

Joint semantic tilting

$$(D^r_\mathbf{x}, D^r_\mathbf{z})(\tilde{\mathbf{x}}, \tau_\mathbf{x}, \tilde{\mathbf{z}}, \tau_\mathbf{z}) = (D_\mathbf{x}, D_\mathbf{z})\big(\tilde{\mathbf{x}}, \tau_\mathbf{x}, \tilde{\mathbf{z}} + \mathbf{a}, \tau_\mathbf{z} + \gamma\big), \tag{8}$$

Sampling with the shifted denoiser targets the joint distribution proportional to $q(\mathbf{x}, \mathbf{z}) \exp r(\mathbf{z})$, whose data marginal is the tilted distribution of Section 2.

The precision shift changes $\tau_\mathbf{z}$ while leaving $\tau_\mathbf{x}$ unchanged, so training must cover different combinations of the two noise levels, which is why we sample them independently. Once trained, the same joint model serves different semantic rewards without reward-specific fine-tuning or additional denoiser evaluations.

3.4 Semantic diffusion priors

Instead of jointly diffusing data and semantic representations, we can use a semantic diffusion prior and a separate conditional generator for the data, i.e., $q(\mathbf{x}, \mathbf{z}) = q(\mathbf{z})q(\mathbf{x} \mid \mathbf{z})$. Because the reward depends only on $\mathbf{z}$, the tilted distribution factorizes as

$$q^r(\mathbf{x}, \mathbf{z}) = q^r(\mathbf{z})q(\mathbf{x} \mid \mathbf{z}),$$

where $q^r(\mathbf{z}) \propto q(\mathbf{z}) \exp r(\mathbf{z})$. Sampling then draws $\mathbf{z} \sim q^r(\mathbf{z})$ and generates data from the unchanged conditional $\mathbf{x} \sim q(\mathbf{x} \mid \mathbf{z})$. If the semantic prior is a diffusion model, its tilted denoiser follows directly from Eq. 7. The conditional generator can be of any type, not necessarily a diffusion. Models such as unCLIP (Ramesh et al., 2022) and Kandinsky (Razzhigaev et al., 2023) have the structure of a semantic diffusion prior and a separate image decoder. Applying the shift only to the prior gives semantic tilting without retraining either component or increasing the number of denoiser evaluations.

4 Experiments

Our experiments illustrate both sampling approaches. We first use direct conditioning with MolGPT to demonstrate that one generator can control molecular properties chosen after training, and we compare its reward–KL trade-off against separately trained property-conditioned models. We then turn to the diffusion approach. On ImageNet, a single joint diffusion model over data and their CLIP embeddings supports tilting while retaining generation quality comparable to REG. On the factorized Kandinsky model, tilting the pretrained semantic prior adds no training or sampling cost.

4.1 Post-hoc property control with MolGPT

We model $q(\mathbf{x} \mid \tilde{\mathbf{z}})$ with MolGPT (Bagal et al., 2021), an autoregressive SMILES generator initialized from the released unconditional GuacaMol checkpoint. CLAMP (Seidl et al., 2023) maps each molecule to a 768-dimensional embedding, which we normalize to obtain $\phi(\mathbf{x})$. During training, for each molecule we sample $(\tilde{\mathbf{z}}, \tau_\mathbf{z})$ according to Eq. 2. Because $\|\phi(\mathbf{x})\| = 1$, the quadratic term in Eq. 3 is constant across molecules, and $q(\mathbf{x} \mid \tilde{\mathbf{z}}, \tau_\mathbf{z}) = q(\mathbf{x} \mid \tilde{\mathbf{z}})$. We therefore condition MolGPT on a fixed rescaling of $\tilde{\mathbf{z}}$, supplied as three prefix tokens, and train it with the standard next-token cross-entropy loss on the SMILES sequence. Implementation details are given in Appendix C.

Independently of generator training, we fit linear predictors of ring count, molecular weight, rotatable bonds, topological polar surface area (TPSA), logarithm of octanol-water partition coefficient (logP), and synthetic accessibility score (SAS) from normalized CLAMP embeddings. For a predictor $\widehat{R}(\mathbf{x}) = \mathbf{w} \cdot \phi(\mathbf{x}) + b$, a reward coefficient c determines the conditioning vector $\tilde{\mathbf{z}} = c\mathbf{w}/s_{\widehat{R}}$, where $s_{\widehat{R}}$ is the standard deviation of the predicted property. The predictors use property labels, but changing the property or c does not require retraining the generator.

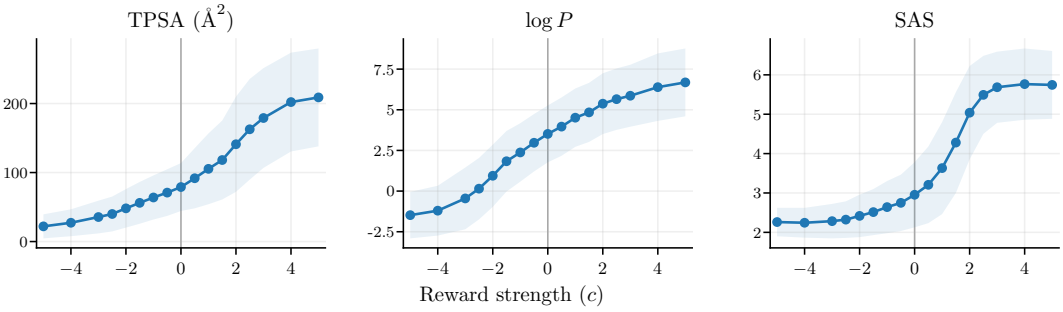

Figure 2: **A single generator controls multiple properties chosen after training.** The generator is trained to condition on noisy CLAMP embeddings without using molecular property values. Property labels are used only to fit the post-hoc linear predictors that define the conditioning vectors. Shaded regions show one standard deviation among valid molecules. More properties in Figure 9.

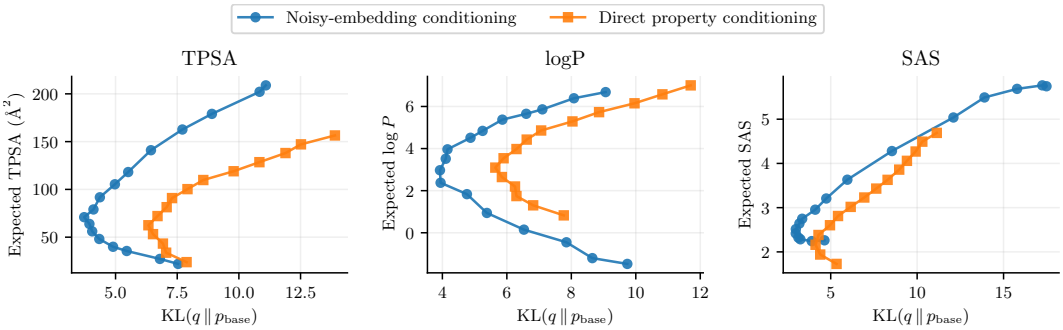

Figure 3: **Noisy-embedding conditioning gives a better reward–KL trade-off than direct property conditioning.** The direct property-conditioning curves are produced by three separate MolGPT models, each trained on one property using the original MolGPT conditioning mechanism. Each panel compares the mean property value with the KL divergence from the same unconditional model.

Figure 2 shows control of multiple molecular properties over $-5 \leq c \leq 5$, with results for more properties in Appendix C. The responses follow a monotone trend across this range. Overall, 96.76% of generations are valid, and 99.38% of valid generations are unique within their condition.

We next compare this model with direct property conditioning. We train three separate MolGPT models, one each for TPSA, logP, and SAS, following the original MolGPT property-conditioning setup. For each property, we compare its mean value with the KL divergence from the same separately trained unconditional reference model. As shown in Figure 3, noisy-embedding conditioning reaches the same property values at substantially lower KL across the evaluated settings.

Simultaneous control of multiple properties. We also demonstrate that multiple properties can be controlled simultaneously for no extra cost. We control logP and SAS jointly using a weighted sum of their standardized predictor directions as the conditioning input. Figure 4 shows that both mean properties move in the requested directions relative to the untilted baseline for each tested combination of reward weights.

4.2 Joint diffusion on ImageNet

We evaluate the joint diffusion construction on 256×256 ImageNet (Deng et al., 2009). We closely follow REG (Wu et al., 2025), a SiT-XL/2 model (Ma et al., 2024) that augments the image tokens with a DINOv2 (Oquab et al., 2024) embedding token. We retain the DINOv2 token and add a normalized CLIP (Radford et al., 2021) embedding token for the rewards considered below. The image and DINOv2 coordinates share a noise level, while the CLIP noise level is sampled independently, as required to tilt the CLIP coordinate without changing the image noise level. We provide a detailed ablation study in Appendix D.

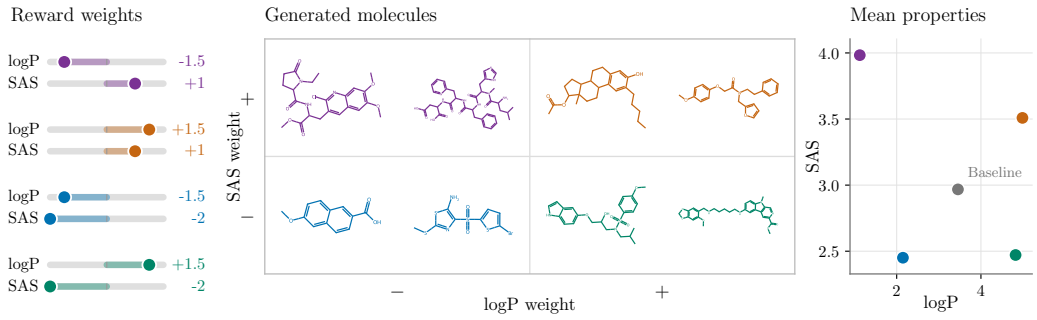

Figure 4: **Simultaneous control of multiple properties.** Left: requested control directions for SAS and logP. Middle: selected generated molecules for each setting. Right: Mean properties in each setting compared to the untilted baseline.

Following the sampling protocol of REG, we use 250 Euler–Maruyama steps to generate 50,000 images and track FID (Heusel et al., 2017) over training (Figure 5). Our model has slightly lower FID through 1M training steps, but then begins to saturate. REG becomes slightly better after roughly 2M steps; at 2.4M steps, FID is 2.319 for our model compared to the reported value of 2.2 for REG.

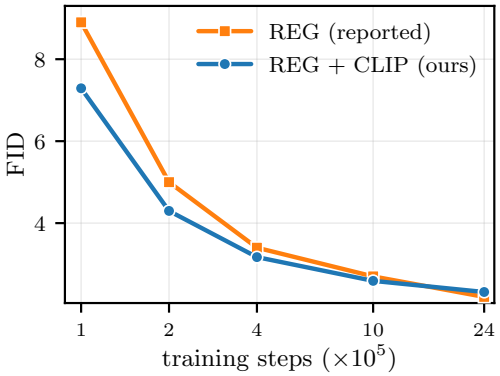

Figure 5: **FID over training.** Our model improves faster early in training, while REG is slightly better after roughly 2M steps.

Prompt-based editing. Given a text prompt, let $\mathbf{z}^\star(\text{prompt})$ denote its CLIP text embedding and define the reward $-\frac{\lambda}{2}\|\mathbf{z} - \mathbf{z}^\star(\text{prompt})\|^2$, corresponding to $\mathbf{a} = \lambda \mathbf{z}^\star(\text{prompt})$ and $\gamma = \lambda$. For a given image $\mathbf{x}$, we compute its CLIP embedding $\mathbf{z} = \phi(\mathbf{x})$ and encode $(\mathbf{x}, \mathbf{z})$ by running the PF-ODE forward using the joint denoiser $(D_\mathbf{x}, D_\mathbf{z})$ to obtain $(\mathbf{x}_T, \mathbf{z}_T)$. We then run the PF-ODE backward using the tilted joint denoiser $(D^r_\mathbf{x}, D^r_\mathbf{z})$. The CLIP similarity between the prompt and the decoded image increases monotonically with λ (Figure 6).

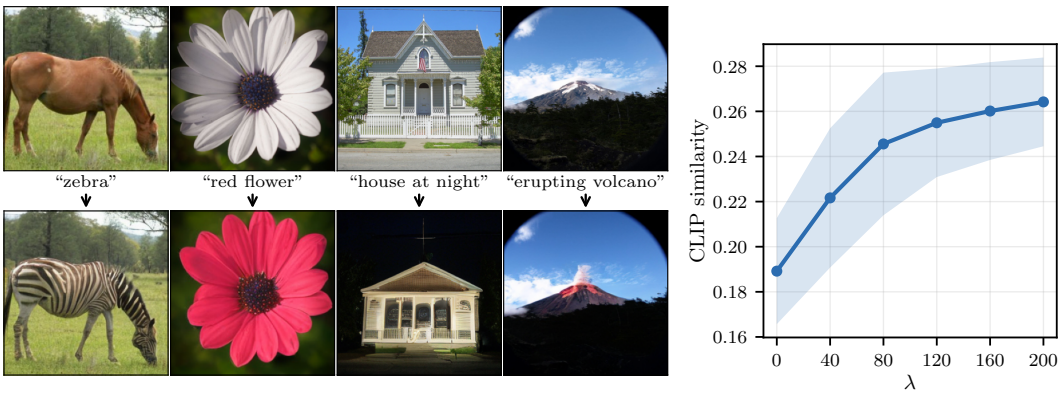

Figure 6: **Prompt-based image editing via semantic tilting.** Source images (top) edited by tilting toward the CLIP text embedding of a prompt (bottom). Right: CLIP similarity between edited images and prompt increases monotonically with λ (mean $\pm$ std over 192 source images).

Aesthetic-score tilting. The LAION aesthetic score predictor (Schuhmann et al., 2022), which estimates human-perceived aesthetic quality, is a linear projection of the normalized CLIP embedding

with weight vector $\mathbf{a}$. We use the reward $r(\mathbf{z}) = \lambda(\mathbf{a} \cdot \mathbf{z} - \frac{\gamma}{2}\|\mathbf{z}\|^2)$, which shifts the embedding coordinates to $(\tilde{\mathbf{z}} + \lambda\mathbf{a},\ \tau_{\mathbf{z}} + \lambda\gamma)$. Here λ controls the overall reward strength and γ the relative quadratic weight. With $\gamma = 0.01$, the aesthetic score increases monotonically with λ (Figure 7). Although, in theory, γ does not change the target distribution for normalized embeddings, it can affect the output of a learned denoiser; we examine this sensitivity in Appendix E.

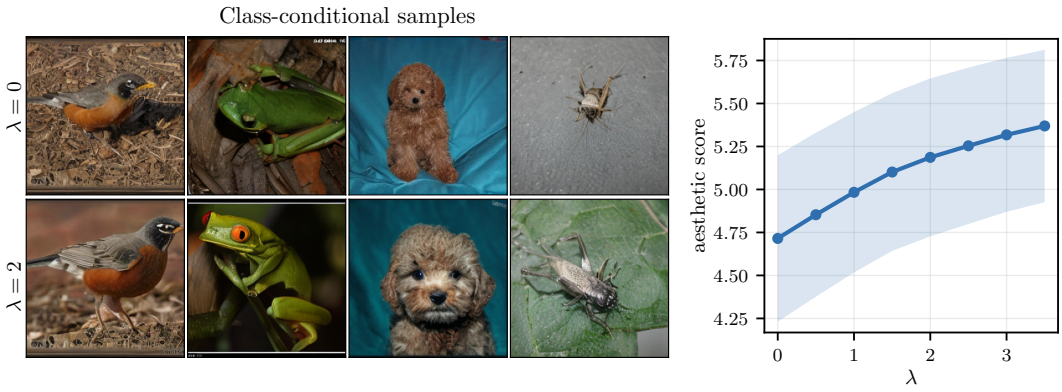

Figure 7: **Aesthetic tilting monotonically improves aesthetic score.** Selected class-conditional ImageNet samples at $\lambda = 0$ (top) and $\lambda = 2$ (bottom). Right: LAION aesthetic score increases monotonically with tilt strength (mean $\pm$ std over 192 samples).

4.3 Tilting a pretrained semantic prior

We next apply the factorized construction to a pretrained model. Kandinsky 2.2 (Razzhigaev et al., 2023) consists of a text-conditional diffusion prior $q(\mathbf{z} \mid \text{prompt})$ and a separate decoder $p(\mathbf{x} \mid \mathbf{z})$. We apply the shifted denoiser from Eq. 7 to the prior and leave the decoder unchanged, without retraining either component.

In Figure 8, we tilt $q(\mathbf{z} \mid \text{prompt})$ toward the embedding $\mathbf{z}^\star = \phi(\mathbf{x}^\star)$ of a reference image $\mathbf{x}^\star$, and compare it to sampling from the untilted model and interpolating between the sampled embedding and $\mathbf{z}^\star$. In this example, tilting incorporates the reference style while retaining the lighthouse scene specified by the prompt, whereas interpolation increasingly reproduces the reference composition.

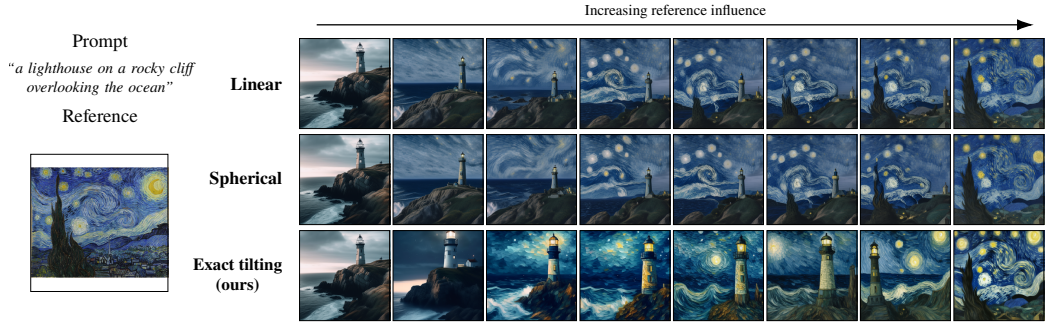

Figure 8: **Semantic tilting on a pretrained factorized model without retraining.** Kandinsky 2.2 samples from a text-conditional prior tilted toward the CLIP embedding of a reference image (Starry Night), compared with linear and spherical interpolation between the sampled and reference embeddings. Tilting gradually incorporates the reference style while maintaining prompt coherence.

5 Conclusion and limitations

We proposed *semantic noising* as a framework for tilting generative models with linear-quadratic rewards in semantic space, and demonstrated its capabilities across molecular and image generation with autoregressive and diffusion models. The framework amortizes the cost of generator training across a family of rewards, allowing a single model to support objectives and strengths chosen after training.

Semantic noising also comes with limitations. First, training joint diffusion models with independent noise levels makes the learning problem harder, and our experiments show that the resulting tilting capabilities come at a small FID cost. Second, theoretically equivalent tilts can produce substantially different results with learned, imperfect models, as demonstrated in our gamma ablation (Appendix E).

AI use statement

In this work, we used generative AI tools to assist with writing experiment code, including implementing methods specified by the authors. We also used these tools to identify relevant literature, brainstorm, draft parts of the paper, and edit the text to improve readability. The authors reviewed all AI-assisted work, checked AI-generated code for correctness, and finalized all manuscript text.

We have *not* used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in writing proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, support qualitative and thematic data analysis, or interpret results. The following tasks are not applicable to this work: generating synthetic datasets, assisting with translation, and cleaning and reformatting datasets.

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

Ethics statement

This paper presents work that advances controllable generative models. While these advancements could enhance applications in creative and scientific domains, they also raise considerations around synthetic media generation and potential misuse. However, our contributions primarily aim at improving understanding and control in generative modeling, without introducing new ethical risks beyond those already associated with generative AI.

Reproducibility statement

We provide experimental details in the appendix. Additionally, we upload code used for all our experiments as part of the submission.

References

Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *ICLR*, 2023.

Viraj Bagal, Rishal Aggarwal, P. K. Vinod, and U. Deva Priyakumar. MolGPT: Molecular generation using a transformer-decoder model. *JCIM*, 2021.

Fan Bao, Shen Nie, Kaiwen Xue, Chongxuan Li, Shi Pu, Yaole Wang, Gang Yue, Yue Cao, Hang Su, and Jun Zhu. One transformer fits all distributions in multi-modal diffusion at scale. In *ICML*, 2023.

Andreas Bergmeister, Stefanie Jegelka, Nikolas Nüsken, Carles Domingo-Enrich, and Jakiw Pidstrigach. Reinforce adjoint matching: Scaling rl post-training of diffusion and flow-matching models, 2026.

Nathan Brown, Marco Fiscato, Marwin H.S. Segler, and Alain C. Vaucher. GuacaMol: Benchmarking models for de novo molecular design. *JCIM*, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

Carles Domingo-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 *ICLR*, 2025.

Monroe D Donsker and SR Srinivasa Varadhan. Asymptotic evaluation of certain markov process expectations for large time. iv. *Communications on pure and applied mathematics*, 1983.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

Ronald Aylmer Fisher. Two new properties of mathematical likelihood. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 1934.

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 *NeurIPS*, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

Peter Holderrieth, Uriel Singer, Tommi Jaakkola, Ricky T. Q. Chen, Yaron Lipman, and Brian Karrer. GLASS flows: Efficient inference for reward alignment of flow and diffusion models. In *ICLR*, 2026.

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *JMLR*, 2005.

Rafał Karczewski, Markus Heinonen, Alison Pouplin, Søren Hauberg, and Vikas K Garg. The spacetime of diffusion models: An information geometry perspective. In *ICLR*, 2026.

Theodoros Kouzelis, Efstathios Karypidis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. Boosting generative image modeling via joint image-feature synthesis. In *NeurIPS*, 2025.

Theodoros Kouzelis, Spyros Gidaris, and Nikos Komodakis. Coevolving representations in joint image-feature diffusion. *arXiv preprint arXiv:2604.17492*, 2026.

G Landrum, 2010. URL https://www.rdkit.org. RDKit: Open-source cheminformatics.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di ZHANG, and Wanli Ouyang. Flow-GRPO: Training flow matching models via online RL. In *NeurIPS*, 2025.

Qiang Liu. Let us flow together, 2024. URL https://www.cs.utexas.edu/~lqiang/PDF/flow_book.pdf. Lecture notes.

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

Cheng Lu, Huayu Chen, Jianfei Chen, Hang Su, Chongxuan Li, and Jun Zhu. Contrastive energy prediction for exact energy-guided diffusion sampling in offline reinforcement learning. In *ICML*, 2023.

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. SiT: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *ECCV*, 2024.

Curtis McDonald. Technical note on relating scores of tilted distributions. *arXiv preprint arXiv:2604.27196*, 2026.

Ankur Moitra, Andrej Risteski, and Dhruv Rohatgi. Steering diffusion models with quadratic rewards: A fine-grained analysis. In *Proceedings of the Thirty-Ninth Conference on Learning Theory*, volume 336 of *Proceedings of Machine Learning Research*, pp. 5188–5209. PMLR, 2026.

Andrea Montanari. Sampling, diffusions, and stochastic localization. *arXiv preprint arXiv:2305.10690*, 2023.

Frank Nielsen and Vincent Garcia. Statistical exponential families: A digest with flash cards. *CORR*, 2009.

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, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *TMLR*, 2024.

Peter Potaptchik, Lee Cheuk Kit, and Michael Samuel Albergo. Tilt matching for scalable sampling and fine-tuning. In *ICML*, 2026.

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 *ICML*, 2021.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*, 2022.

Anton Razzhigaev, Arseniy Shakhmatov, Anastasia Maltseva, Vladimir Arkhipkin, Igor Pavlov, Ilya Ryabov, Angelina Kuts, Alexander Panchenko, Andrey Kuznetsov, and Denis Dimitrov. Kandinsky: an improved text-to-image synthesis with image prior and latent diffusion. In *EMNLP: System Demonstrations*, 2023.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R 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 *NeurIPS Datasets and Benchmarks Track*, 2022.

Philipp Seidl, Andreu Vall, Sepp Hochreiter, and Günter Klambauer. Enhancing activity prediction models in drug discovery with the ability to understand human language. In *ICML*, 2023.

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 *ICML*, 2025.

Marta Skreta, Tara Akhound-Sadegh, Viktor Ohanesian, Roberto Bondesan, Alan Aspuru-Guzik, Arnaud Doucet, Rob Brekelmans, Alexander Tong, and Kirill Neklyudov. Feynman-kac correctors in diffusion: Annealing, guidance, and product of experts. In *ICML*, 2025.

Yang Song, Jascha Sohl-Dickstein, Diederik Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 2011.

Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, Zhaowei Chen, Hongcheng Gao, Yao Tang, Jian Yang, Ming-Ming Cheng, and Xiang Li. Representation entanglement for generation: Training diffusion transformers is much easier than you think. In *NeurIPS*, 2025.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *ICLR*, 2025.

A EXPONENTIAL FAMILIES

A family of distributions $\{p(\mathbf{x} \mid \boldsymbol{\theta})\}_{\boldsymbol{\theta}}$ parametrized by $\boldsymbol{\theta} \in \mathbb{R}^k$ is a k-dimensional exponential family if its density can be written as (Fisher, 1934; Nielsen & Garcia, 2009)

$$p(\mathbf{x} \mid \boldsymbol{\theta}) \propto h(\mathbf{x}) \exp(\boldsymbol{\eta}(\boldsymbol{\theta}) \cdot T(\mathbf{x})), \tag{9}$$

where h is the base measure, $\boldsymbol{\eta}(\boldsymbol{\theta})$ is the *natural parameter*, and $T(\mathbf{x})$ is the *sufficient statistic*. Since the distribution depends on $\boldsymbol{\theta}$ only through $\boldsymbol{\eta}(\boldsymbol{\theta})$, we can instead parametrize the family directly by its natural parameter:

$$\pi(\mathbf{x} \mid \boldsymbol{\eta}) \propto h(\mathbf{x}) \exp(\boldsymbol{\eta} \cdot T(\mathbf{x})). \tag{10}$$

Then $p(\mathbf{x} \mid \boldsymbol{\theta}) = \pi(\mathbf{x} \mid \boldsymbol{\eta}(\boldsymbol{\theta}))$.

A.1 TILTING WITH REWARDS LINEAR IN T

For a reward $r(\mathbf{x}) = \mathbf{a} \cdot T(\mathbf{x})$, tilting π gives

$$\begin{aligned} \pi^r(\mathbf{x} \mid \boldsymbol{\eta}) &\propto h(\mathbf{x}) \exp(\boldsymbol{\eta} \cdot T(\mathbf{x})) \exp(\mathbf{a} \cdot T(\mathbf{x})) = h(\mathbf{x}) \exp((\boldsymbol{\eta} + \mathbf{a}) \cdot T(\mathbf{x})) \\ &\propto \pi(\mathbf{x} \mid \boldsymbol{\eta} + \mathbf{a}). \end{aligned} \tag{11}$$

Thus, exponential families are closed under rewards linear in the sufficient statistic: tilting translates the natural parameter from $\boldsymbol{\eta}$ to $\boldsymbol{\eta} + \mathbf{a}$.

B DENOISING POSTERIORS FORM AN EXPONENTIAL FAMILY

The denoising posterior $p_t(\mathbf{x} \mid \mathbf{x}_t)$ is the distribution of clean data given a noisy observation $\mathbf{x}_t$. By Bayes' rule,

$$p_t(\mathbf{x} \mid \mathbf{x}_t) \propto q(\mathbf{x}) \exp\left(\frac{\alpha_t \mathbf{x}_t}{\sigma_t^2} \cdot \mathbf{x} - \frac{\alpha_t^2}{2\sigma_t^2} \|\mathbf{x}\|^2\right). \tag{12}$$

Therefore, the denoising posteriors form an $(n+1)$-dimensional exponential family with base measure q, sufficient statistic

$$T(\mathbf{x}) = \left(\mathbf{x}, -\tfrac{1}{2}\|\mathbf{x}\|^2\right), \tag{13}$$

and natural parameter (Karczewski et al., 2026)

$$\boldsymbol{\eta}(\mathbf{x}_t, t) = \left(\frac{\alpha_t \mathbf{x}_t}{\sigma_t^2}, \frac{\alpha_t^2}{\sigma_t^2}\right) = (\tilde{\mathbf{x}}, \tau). \tag{14}$$

By expressing the denoising posterior directly in the natural parametrization, we have

$$p_t(\mathbf{x} \mid \mathbf{x}_t) = \pi(\mathbf{x} \mid \boldsymbol{\eta}(\mathbf{x}_t, t)). \tag{15}$$

The denoiser in natural coordinates is

$$D(\boldsymbol{\eta}) = \int \mathbf{x}\, \pi(\mathbf{x} \mid \boldsymbol{\eta}) d\mathbf{x}, \tag{16}$$

and is related to the schedule-dependent denoiser by

$$D_{(\alpha,\sigma)}(\mathbf{x}_t, t) = D(\boldsymbol{\eta}(\mathbf{x}_t, t)) = D\left(\frac{\alpha_t \mathbf{x}_t}{\sigma_t^2}, \frac{\alpha_t^2}{\sigma_t^2}\right). \tag{17}$$

B.1 LINEAR-QUADRATIC TILTING IN NATURAL COORDINATES

Consider the diffusion process obtained by applying the noising kernel to the tilted distribution $q^r \propto q \exp(r)$. The resulting denoising posterior is

$$\pi^r(\mathbf{x} \mid \boldsymbol{\eta}) \propto q^r(\mathbf{x}) \exp(\boldsymbol{\eta} \cdot T(\mathbf{x})) \propto \pi(\mathbf{x} \mid \boldsymbol{\eta}) \exp(r(\mathbf{x})). \tag{18}$$

The corresponding denoiser is

$$D^r(\boldsymbol{\eta}) = \frac{\mathbb{E}[\mathbf{x} \exp(r(\mathbf{x})) \mid \boldsymbol{\eta}]}{\mathbb{E}[\exp(r(\mathbf{x})) \mid \boldsymbol{\eta}]}, \tag{19}$$

which is generally intractable. However, for

$$r(\mathbf{x}) = \mathbf{a} \cdot \mathbf{x} - \tfrac{\gamma}{2}\|\mathbf{x}\|^2 = (\mathbf{a}, \gamma) \cdot T(\mathbf{x}), \tag{20}$$

the reward is linear in the sufficient statistic. Applying Eq. 11 gives

$$\pi^r(\mathbf{x} \mid (\tilde{\mathbf{x}}, \tau)) = \pi(\mathbf{x} \mid (\tilde{\mathbf{x}} + \mathbf{a}, \tau + \gamma)), \tag{21}$$

and hence

$$D^r(\tilde{\mathbf{x}}, \tau) = D(\tilde{\mathbf{x}} + \mathbf{a}, \tau + \gamma). \tag{22}$$

The purely linear case sets $\gamma = 0$ (Moitra et al., 2026). The quadratic proximity reward sets $\mathbf{a} = \lambda \mathbf{x}^\star$ and $\gamma = \lambda$ (Liu, 2024).

Posterior sampling as a special case. Let $\boldsymbol{\eta}^\star = \boldsymbol{\eta}(\mathbf{x}_t, t)$ be the natural parameter of a fixed noisy observation $\mathbf{x}_t$ at diffusion time t. By Eq. 12, its posterior is q tilted by $\boldsymbol{\eta}^\star \cdot T(\mathbf{x})$. Sampling from this posterior with the base diffusion model therefore uses

$$D_{\text{posterior}}(\boldsymbol{\eta}) = D(\boldsymbol{\eta} + \boldsymbol{\eta}^\star). \tag{23}$$

This is the posterior-sampling formula of Holderrieth et al. (2026) expressed in natural coordinates.

B.2 Joint linear-quadratic tilting

Suppose that $\mathbf{x}$ and $\mathbf{z}$ are noised independently. The joint denoising posterior is an exponential family with natural parameter $\boldsymbol{\eta} = (\tilde{\mathbf{x}}, \tau_{\mathbf{x}}, \tilde{\mathbf{z}}, \tau_{\mathbf{z}})$ and sufficient statistic $T(\mathbf{x}, \mathbf{z}) = (\mathbf{x}, -\frac{1}{2}\|\mathbf{x}\|^2, \mathbf{z}, -\frac{1}{2}\|\mathbf{z}\|^2)$:

$$\pi(\mathbf{x}, \mathbf{z} \mid \boldsymbol{\eta}) \propto q(\mathbf{x}, \mathbf{z}) \exp\big(\boldsymbol{\eta} \cdot T(\mathbf{x}, \mathbf{z})\big). \tag{24}$$

A linear-quadratic reward that depends only on the embedding is linear in the $\mathbf{z}$-components of T:

$$r(\mathbf{z}) = \mathbf{a} \cdot \mathbf{z} - \tfrac{\gamma}{2}\|\mathbf{z}\|^2 = (\mathbf{0}, 0, \mathbf{a}, \gamma) \cdot T(\mathbf{x}, \mathbf{z}). \tag{25}$$

By Eq. 11, the tilted posterior has shifted natural parameter

$$\pi^r(\mathbf{x}, \mathbf{z} \mid (\tilde{\mathbf{x}}, \tau_{\mathbf{x}}, \tilde{\mathbf{z}}, \tau_{\mathbf{z}})) = \pi(\mathbf{x}, \mathbf{z} \mid (\tilde{\mathbf{x}}, \tau_{\mathbf{x}}, \tilde{\mathbf{z}} + \mathbf{a}, \tau_{\mathbf{z}} + \gamma)). \tag{26}$$

Taking marginal expectations gives the joint tilted denoisers:

$$(D^r_{\mathbf{x}}, D^r_{\mathbf{z}})(\tilde{\mathbf{x}}, \tau_{\mathbf{x}}, \tilde{\mathbf{z}}, \tau_{\mathbf{z}}) = (D_{\mathbf{x}}, D_{\mathbf{z}})(\tilde{\mathbf{x}}, \tau_{\mathbf{x}}, \tilde{\mathbf{z}} + \mathbf{a}, \tau_{\mathbf{z}} + \gamma). \tag{27}$$

Only the $(\tilde{\mathbf{z}}, \tau_{\mathbf{z}})$ coordinates are shifted; the state coordinates $(\tilde{\mathbf{x}}, \tau_{\mathbf{x}})$ are unchanged.

C MolGPT experimental details

We train five MolGPT models for the experiments in Section 4.1: the noisy-embedding model, three direct property-conditioned models, and an unconditional reference model. Each is initialized independently from the released unconditional GuacaMol checkpoint (Bagal et al., 2021). The architecture has eight transformer layers, eight attention heads, width 256, and a 100-token context.

All five models use the same dataset of 1,273,090 GuacaMol molecules (Brown et al., 2019), shuffled order, and training schedule. We train for 100 epochs (331,600 updates) with batch size 384 and teacher-forced next-token prediction. We use AdamW with coefficients $(0.9, 0.95)$, weight decay 0.1, and gradient clipping at 1. The learning rate warms up to 10^{-4} over 335 updates and then follows cosine decay, clipped at 10^{-5}. The input token embeddings remain frozen.

Noisy-embedding conditioning. The frozen CLAMP molecular encoder (Seidl et al., 2023) produces 768-dimensional features from Morgan fingerprint counts, which we normalize to obtain $\boldsymbol{\phi}(\mathbf{x})$. During training, we set $\tau_{\mathbf{z}} = 0$ with probability 0.05 and otherwise sample $\sqrt{\tau_{\mathbf{z}}} \sim \text{Uniform}[0.1, 32]$. We then sample

$$\tilde{\mathbf{z}} = \tau_{\mathbf{z}} \boldsymbol{\phi}(\mathbf{x}) + \sqrt{\tau_{\mathbf{z}}}\boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \tag{28}$$

as in Eq. 2. Since $\boldsymbol{\phi}(\mathbf{x})$ is normalized, the posterior does not depend on $\tau_{\mathbf{z}}$ after $\tilde{\mathbf{z}}$ is fixed. We pass $\tilde{\mathbf{z}}/(32\sqrt{768 + 32^2})$ through a 768-to-768 MLP and split the output into three prefix tokens.

Direct property conditioning. We train three independent models, one each for TPSA, logP, and SAS, following the original MolGPT property-conditioning setup. Each model maps its standardized scalar property through the learned `prop_nn` module to a property prefix token.

Unconditional reference model. The unconditional model is trained with a constant prefix token. We use this model as a reference to estimate KL for the noisy-embedding and direct property-conditioned models.

Property directions. Molecular property values are not used to train the noisy-embedding generator. We independently fit linear predictors on normalized CLAMP embeddings. For a predictor $\widehat{R}(\mathbf{x}) = \mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{x}) + b$, we define the reward

$$r_c(\mathbf{x}) = c\frac{\widehat{R}(\mathbf{x}) - \mu_{\widehat{R}}}{s_{\widehat{R}}} = \tilde{\mathbf{z}} \cdot \boldsymbol{\phi}(\mathbf{x}) + \text{const}, \qquad \tilde{\mathbf{z}} = \frac{c}{s_{\widehat{R}}}\mathbf{w}. \tag{29}$$

At inference time, we supply this $\tilde{\mathbf{z}}$ directly, using the same fixed reference scaling as in training. The quadratic term is constant because $\|\boldsymbol{\phi}(\mathbf{x})\| = 1$, so $\tau_{\mathbf{z}}$ need not be specified separately. At $c = 0$, the conditioning input is exactly $\tilde{\mathbf{z}} = \mathbf{0}$.

Table 1: **Linear property predictors used for MolGPT guidance.** Predictor quality is evaluated on 238,632 held-out molecules. The final column gives the norm of the $|c| = 5$ query after division by the training reference norm $32\sqrt{768 + 32^2}$.

Property	Held-out R^2	Scaled $\|\tilde{\mathbf{z}}\|$ at $\|c\| = 5$
Ring count	0.780	0.615
Molecular weight	0.723	0.520
Rotatable bonds	0.585	0.761
TPSA	0.750	0.648
logP	0.778	0.565
SAS	0.798	0.607

We selected ring count, molecular weight, and rotatable bonds from six additional candidates before inspecting their generation results. Ring count and molecular weight were predicted accurately from CLAMP; we included rotatable-bond count because it measures molecular flexibility and is nearly uncorrelated with ring count. We omitted hydrogen-bond donor and acceptor counts because both correlated strongly with TPSA; QED was predicted less accurately.

Evaluation. For each condition, we generate 2,048 sequences from the initial token `C`, followed by at most 100 additional tokens, using temperature one and no top-k restriction. RDKit (Landrum, 2010) determines validity and evaluates the displayed properties.

For Figure 3, we estimate

$$D_{\mathrm{KL}}(q\|p_{\mathrm{base}}) = \mathbb{E}_{\mathbf{x}\sim q}\left[\log q(\mathbf{x}) - \log p_{\mathrm{base}}(\mathbf{x})\right] \tag{30}$$

using the exact autoregressive probability of each sampled sequence. Invalid sequences remain in the KL estimate, while property means are computed over valid molecules. For the noisy-embedding model, we use $c \in \{0, \pm 0.5, \pm 1, \pm 1.5, \pm 2, \pm 2.5, \pm 3, \pm 4, \pm 5\}$. For each direct property-conditioned model, the standardized target values range from -1.5 to 2.0 in increments of 0.25.

For Figure 9, the same grid of nonzero values of c is used for each of the six properties, together with one shared $\tilde{\mathbf{z}} = \mathbf{0}$ distribution. This gives 97 conditions and 198,656 attempted generations. Of these, 192,229 are valid (96.76%), and 191,042 are unique within their condition (99.38% of valid generations). The shaded regions show the mean plus or minus one population standard deviation among valid molecules within each condition.

D JOINT DIFFUSION MODEL ABLATION STUDY

REG (Wu et al., 2025) jointly diffuses image latent tokens and an extra embedding token extracted from a pretrained image encoder. It also uses REPA (Yu et al., 2025) to align intermediate denoiser

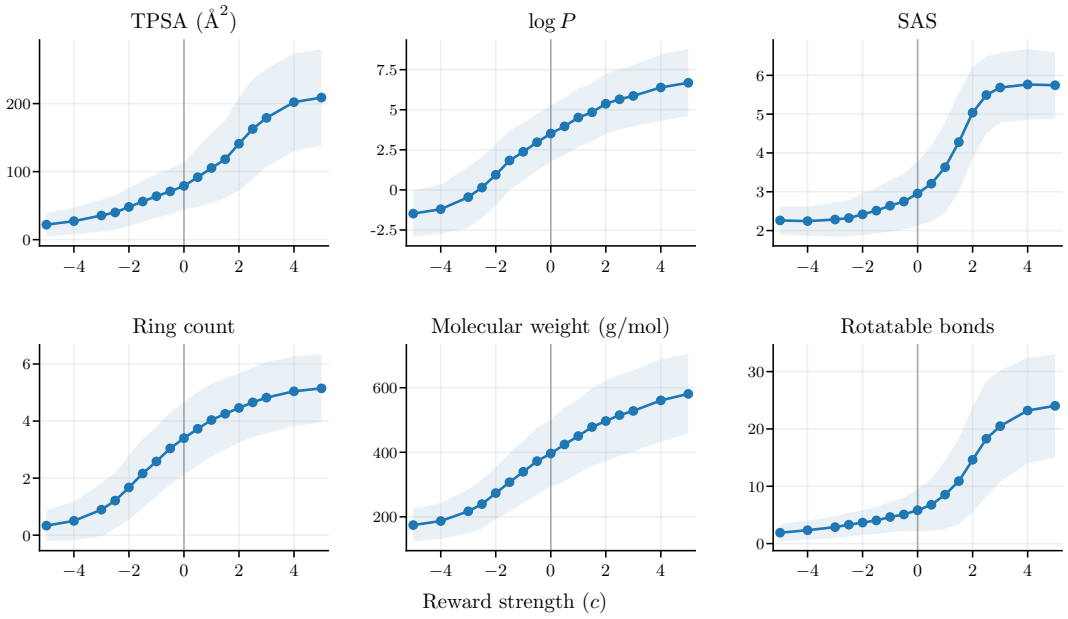

Figure 9: **Single-property control with the same noisy-embedding generator.** The full set of results includes ring count, molecular weight, and rotatable bonds in addition to the properties shown in Figure 2. Each panel varies the reward coefficient for one property and reports the resulting RDKit property values; shaded regions show one standard deviation among valid molecules.

patch representations with those of the same encoder during training. The original study reports that DINOv2 (Oquab et al., 2024) outperforms CLIP (Radford et al., 2021) for generation quality when used as both the REPA target and the generated embedding. Our downstream applications nevertheless require a CLIP coordinate: text-based editing uses CLIP's text embedding, and the AVA aesthetic predictor (Schuhmann et al., 2022) is a linear projection of its image embedding. We therefore study how to add a CLIP coordinate without sacrificing generation quality.

We reproduced REG at B/2 scale as a calibration baseline (FID 15.50 at 400K training steps vs. reported 15.22) and conducted all ablations at this scale before scaling the selected recipe to XL/2.

Metrics. We report FID and CLIP consistency. FID is computed from 50K class-conditioned samples without classifier-free guidance; lower is better. To measure CLIP consistency, we fix a clean CLIP embedding as the conditioning input ($t_{\mathbf{z}} = 0$), generate an image with the usual image and DINO noise schedule, re-encode it with CLIP, and compute its cosine similarity with the conditioning embedding. We average this similarity over 1,024 samples; higher is better.

D.1 Representation choice

Our first model replaced DINOv2 with CLIP throughout: CLIP patch tokens as REPA targets and a normalized CLIP global embedding as the generated coordinate. This confirmed REG's finding: FID at 100K steps was 43.27, compared with 35.39 for our REG reproduction at the same step.

The REPA patch targets are used only during training, whereas the global coordinate is jointly diffused with the image and generated at inference time. We varied these choices separately across five B/2 configurations with shared noise levels.

Patch teacher. Switching the REPA targets from CLIP to DINOv2 patches, while keeping the CLIP global coordinate unchanged, improves FID at 100K steps from 43.27 to 38.31.

Global coordinate. Further replacing the CLIP global coordinate with DINO CLS improves FID at 100K steps by another 3.01 (from 38.31 to 35.30). Including the CLS token in the REPA average, as the original REG does, has negligible effect (35.30 vs. 35.39).

Dual DINO+CLIP coordinates. Rather than choosing between DINO and CLIP, we keep the DINO CLS token as the primary generated coordinate and add a second generated token trained to predict the normalized CLIP embedding of the clean image. This token has its own input projection, layer normalization, and output head. The addition increases parameter count by 0.85% and introduces a 0.03-weighted CLIP velocity loss. The dual model achieves the best B/2 FID at both 100K steps (33.05) and 400K steps (13.75), improving over REG (15.50) by 11.3%. We retain CLIP as the tilting target because it provides a corresponding text embedding. The DINO noise level could also be sampled independently, which would allow tilting in both semantic spaces; we did not pursue this.

D.2 Noise schedule

During standard generation, all coordinates use the same noise level. Training the dual model with shared noise gives its best FID (13.75 at 400K steps), surpassing REG (15.50), and would therefore be preferable for generation quality alone. Embedding tilting, however, changes the CLIP precision without changing the image precision (Eq. 8). A shared-noise model is not trained with such pairs of noise levels; in particular, it never sees a clean CLIP embedding together with a noisy image.

We explored seven variants, summarized below and in Figure 10. We use the shorthand $(t_{\mathbf{x}}, t_{\mathrm{DINO}}, t_{\mathbf{z}})$ for the noise levels of the image, DINO, and CLIP coordinates, respectively, where t denotes the shared diffusion time and $U_1, U_2, U_3 \sim \mathrm{Uniform}[0, T]$ are independent draws.

V1 DINO+CLIP, all noise levels shared (t, t, t), one shared timestep MLP. Best FID (**13.75**) but weakest CLIP consistency (0.637).

V2 Same trained model as V1, different evaluation: the clean CLIP vector is kept fixed but its timestep input is set to $t_{\mathbf{x}}$ rather than 0. CLIP consistency improves to 0.655.

V3 CLIP only (t, t), shared noise and shared timestep MLP. This is the single-CLIP-coordinate model from Section D.1. FID 17.61, CLIP consistency 0.657.

V4 Same trained model as V3, different evaluation (timestep input set to $t_{\mathbf{x}}$). CLIP consistency improves to 0.697.

V5 DINO+CLIP, all noise levels independent (U_1, U_2, U_3), one shared timestep MLP. FID 18.58, CLIP consistency 0.725.

V6 DINO+CLIP, all noise levels independent (U_1, U_2, U_3), three separate timestep MLPs. FID 16.92, CLIP consistency 0.729.

V7 DINO+CLIP, image and DINO share noise level, CLIP independent (U_1, U_1, U_2), with a separate CLIP timestep MLP. FID 15.40, CLIP consistency **0.736**.

REG baseline (DINO only, no CLIP coordinate): FID 15.50.

Among the shared-noise models, V1 gives the best FID but weakest CLIP consistency. Models trained with independent CLIP noise (V5–V7) achieve substantially higher CLIP consistency.

Within the independent-noise variants, V6 improves over V5 after introducing separate timestep MLPs. V7 couples image and DINO noise (since these are not tilted) while keeping CLIP independent, and achieves the best trade-off: FID approximately matches REG while CLIP consistency is the highest. We selected V7 for scaling to XL. Two additional variants ($t_{\mathbf{z}} = \min(U_1, t_{\mathbf{x}})$ and $t_{\mathbf{z}} = U_1 \cdot t_{\mathbf{x}}$) were stopped at 100K steps due to poor early FID.

D.3 Scaling to XL and practical overhead

We scaled the selected V7 recipe to SiT-XL/2 (688M parameters). Without caching, the dominant additional cost is the CLIP encoder forward pass used to compute training targets. Since the training crops are fixed, we precomputed and cached all CLIP global embeddings; this took roughly 3 minutes on four GH200 GPUs. The remaining overheads relative to REG are:

- **Parameters:** +3.40M (0.50% over REG XL);
- **Training:** +0.26% per-update cost with cached CLIP targets;
- **Inference:** +0.78% per denoiser forward pass from the additional token.

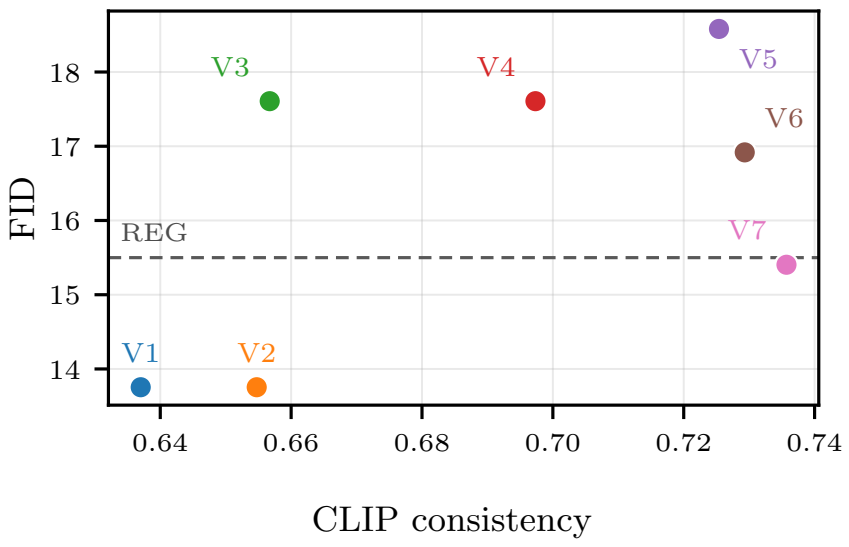

Figure 10: **Generation quality vs. embedding consistency at B/2 scale.** Each point is evaluated at 400K training steps; lower-right is preferable. V1 (shared noise) has the best FID but weakest CLIP consistency. V7 (shared image/DINO noise, independent CLIP noise) achieves the strongest CLIP consistency while approximately matching REG's FID (dashed line). V2 and V4 use the V1 and V3 models but set the CLIP timestep input to $t_{\mathbf{x}}$ rather than 0.

At 800K training steps, the XL model reaches FID 2.66 and CLIP consistency 0.82 without classifier-free guidance. At 2.4M steps, FID improves to 2.319 and CLIP consistency to 0.833, while REG reports FID 2.2.

E Quadratic coefficient ablation

For normalized CLIP embeddings, changing γ does not change the target distribution because $\|\mathbf{z}\|^2$ is constant (Eq. 3). It does, however, change the inputs at which the learned denoiser is evaluated. Figure 11 shows the resulting variation at a fixed reward strength. We use $\gamma = 0.01$ for the aesthetic-score experiments.

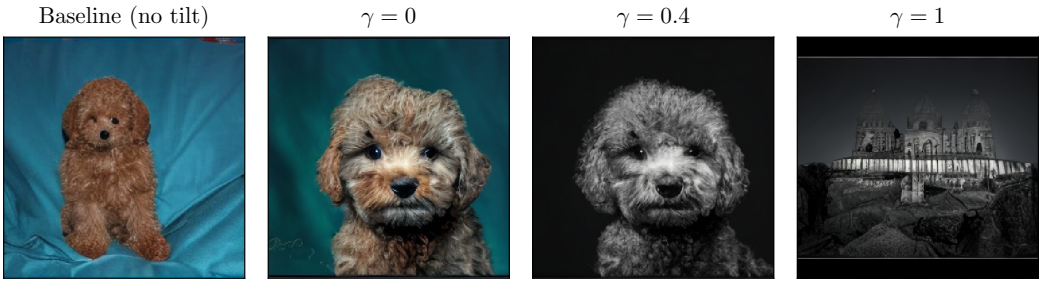

Figure 11: **The learned model depends on γ despite equivalence of the target distributions.** Aesthetic tilting at $\lambda = 3$ with normalized CLIP embeddings and different values of γ.