What Is It Like to Be a Noise?
An Entropy-based Gaussian Noise Regularization for Diffusion Models

Pascal Chang[1,2] Kai Lascheit[1,2] Jingwei Tang[2] Markus Gross[1,2] Vinicius Azevedo[2]

[1]ETH Zürich [2]DisneyResearch|Studios

Abstract

Inference-time optimization of diffusion latents enables powerful control but often degrades the statistical structure of true Gaussian noise, causing artifacts and reward hacking. To address this, we propose a Gaussianity regularizer that aligns a sample's local statistics with a typical Gaussian realization, rather than relying on pointwise likelihood. We formalize this by computing the KL divergence between the sample distribution and the Gaussian prior. To make the divergence computation tractable from a single sample, we lift each candidate latent into an empirical distribution induced by its statistics and model it as a pairwise Markov Random Field. This yields a Bethe–Kikuchi-style regularizer with 1D marginal, 2D spatial, and multi-scale terms. Our results show improved latent optimization stability and generation quality over prior approaches.

1. Introduction

"What is it like to be a noise?" asked the researcher. The noise, momentarily interrupted from its otherwise fulfilling duty of becoming a suspiciously photogenic cat, offered no reply.[1] Yet standard image diffusion models crucially depend on such objects: high-dimensional Gaussian samples serve as a canonical, maximum-entropy starting point, providing an uninformative prior from which data distributions can be learned. Given the success of these models, many recent inference-time methods attempt to repurpose diffusion pipelines for post-hoc objectives such as reward guidance, improved generation quality, and controllable synthesis [4, 19, 21, 38, 44, 57]. A common strategy is to directly optimize the initial noise latent so that the final denoised sample better satisfies the desired objective. However, once this optimization pushes the latent away from the statistical character of true Gaussian noise, the model is forced to denoise samples it was never trained to see, often resulting in artifacts, brittle behavior, and reward hacking [13, 57, 69].

This failure mode turns regularization of optimized latents into a balancing act: the sample should preserve the objective-induced changes while still remaining a valid noise realization. We therefore formulate this task as a multi-objective problem: given a modified latent, the goal is to find a nearby solution that remains faithful to the target objective while also being Gaussian. Measuring Gaussianity, however, is itself a nontrivial problem. In principle, the natural measure is the KL divergence $D_{\mathrm{KL}}(P \,\|\, G)$ between the unknown data distribution P and the Gaussian prior G. We refer to computing this quantity as the *hard problem of Gaussianity*: it captures exactly what we would like to measure, yet is unavailable in practice because P is unknown and cannot be inferred from a single sample.

Our first contribution addresses this intractability by lifting each candidate sample into a distribution induced by the statistics it carries, shifting the question from whether a point "belongs" to the Gaussian prior to whether its local statistics are consistent with those of a typical Gaussian draw. Our second contribution is to model the sample as a pairwise Markov Random Field with local connectivity, which admits a Bethe–Kikuchi-style approximation of the KL objective. Finally, because local pairwise models do not directly constrain longer-range correlations, we apply the loss across a pyramid of downsampled samples, exposing larger-scale structure at coarser resolutions. Our results show that this formulation yields a principled and practical Gaussianity regularizer, improving the stability of latent optimization and outperforming previous approaches.

2. Related Work

Normality tests. The Gaussian distribution, a maximum-entropy law under fixed mean and variance, underlies a vast range of natural and computational processes: from thermal noise [41] and sensor statistics [43] to latent spaces in generative models [27]. Understanding whether a given high-dimensional sample adheres to Gaussian statistics has therefore long motivated the development of normality tests.

[1]We take its silence on the matter as sufficient motivation, and this paper is, in part, an attempt to answer on its behalf.

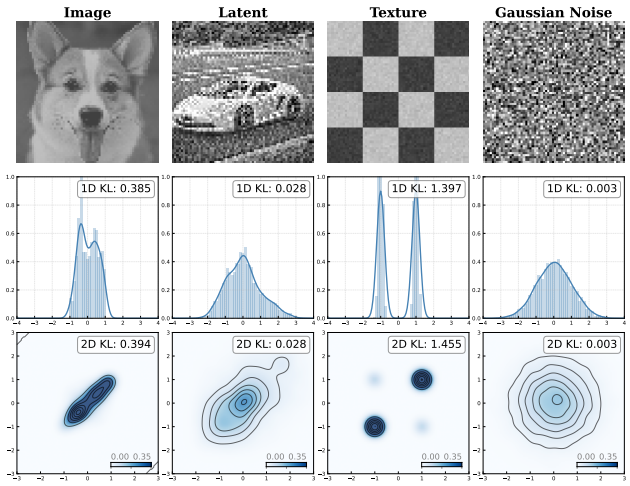

Figure 1. **Value and Spatial Entropy Visualization**. We show the 1D and 2D entropy as estimated with KDE for four different common inputs: image scaled to [-1,1], latent vector, a checkerboard texture, and a Gaussian noise.

Kolmogorov-Smirnov [2], Anderson-Darling [3], Shapiro-Wilk [55] and Jarque-Bera [22] evaluate one-dimensional Gaussianity by comparing marginal distributions through CDF alignment, quantile correlation, or moment matching. While effective for i.i.d. samples, they assume independence and therefore ignore higher-order or spatial correlations, which are central to the structure of diffusion noise. Other works such as Gaussianization [10] propose an iterative dataset transform that alternates between global rotations and marginal cumulative distribution function (CDF) matching to map an entire distribution into approximately independent Gaussian coordinates. Closer to the machine learning community, normality is enforced through distributional alignment losses based on the KL-divergence [27] or kurtosis [56]. Concurrent to our work, Hwang et al. [21] proposes a Gaussianity measure based on moment matching in both the spatial and spectral domain.

Gradient-based noise optimization. A growing body of work [1, 4, 9, 11, 13, 19, 38, 46, 57, 59, 62] explores direct noise optimization, where noise samples of a diffusion model are refined through gradient descent to steer the generative trajectory without retraining or fine-tuning the model. Differentiable image-space rewards [14, 57] such as brightness matching or improving aesthetics can guide the diffusion path toward novel objectives not considered during training, while non-differentiable rewards [9, 38, 67] address discrete goals like quantity-aware or semantic consistency. Unconstrained gradient updates from differentiable rewards quickly drive the optimized noise away from the Gaussian prior, destabilizing the generation process. To mitigate this issue, prior works introduce various Gaussianity-preserving regularizers, including norm-aware constraints [4, 13, 50], noise averaging with new Gaussian samples [59], moment matching on local image subsets [57], and distributional alignment losses based on the KL-divergence [14, 19]. Beyond generating images, the same principle of optimizing noise samples was also explored in other domains such as human motion [23, 70] and interaction [49], and music generation [40]; these examples illustrate the generality of noise-space optimization as a form of differentiable control across modalities.

Seed selection & noise sampling. While gradient-based approaches explicitly modify the initial noise through back-propagation, sampling-based noise optimization/selection methods [14, 26, 34, 35, 51, 60, 67, 69, 71] explore the noise space implicitly by evaluating, selecting, or iteratively resampling candidate seeds. This strategy avoids the risk of gradient drift away from the Gaussian prior, thereby eliminating the need for explicit Gaussianity regularization; its main trade-off though is a higher computational expense and limited precision, since only a very limited subset of the noise space can be explored through sampling. These works often target sampling images with higher aesthetic score [14, 63, 67, 71], image-space feature matching [60], image-to-image translation [18], sampling of images with rare concepts [51], improving inpainting [34, 63], enhancing compositional generation [30, 36, 37] and improving alignment in video diffusion models [26].

Noise manipulation for diverse applications. Several works have exploited the flexibility of noise design in diffusion models to improve generation quality, structure preservation, and controllability [7, 8, 47, 61, 66, 68]. Blue noise can be used instead of white noise in diffusion training [20]: the spatial correlation in blue noise improves generation quality over the standard Gaussian-noise baseline. Similarly, [58] propose an edge-preserving noise scheduling that adds less noise to edge regions and more to smooth regions during training, which results in faster training convergence. Noise is also important when considering video diffusion models [16, 47]. Noise warping [8] derives a new transport operator that is able to maintain the intrinsic properties of the Gaussian noise, and this approach was used to solve video inverse problems [12], improve 3D consistency in score distillation sampling [29, 64], to improve training [33], controllability [7, 28], and to enforce temporal coherency [42] of video diffusion models. Our work complements these previous approaches by providing a principled Gaussianity metric and projection framework, shedding light on the fundamental properties of high-dimensional Gaussian noise samples.

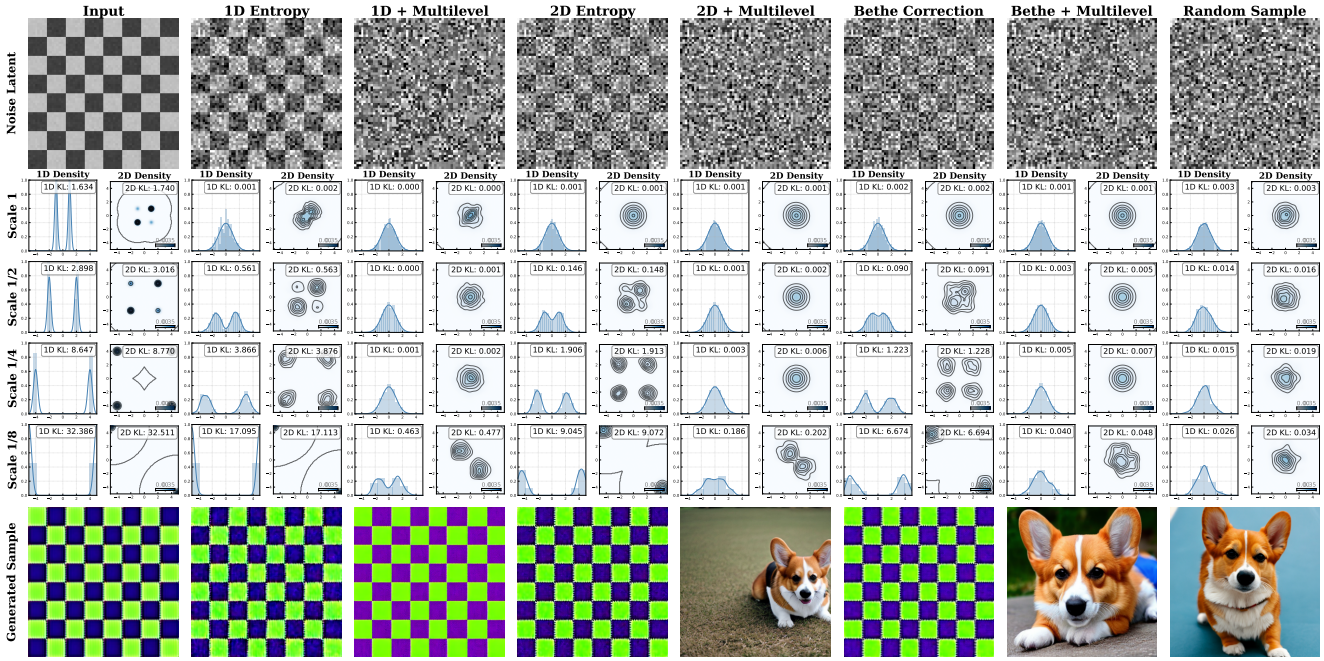

Figure 2. **Effectiveness of Multilevel Bethe Entropy Regularization.** We show the effect of our loss components on the optimized latent (top), its multiscale 1D/2D densities (middle), and the diffusion-generated image (bottom, *A photo of a corgi*). With this checkerboard pattern and learning rate ($\eta = 0.05$), we show that matching first-order statistics alone (using 1D Entropy and 1D Multilevel) cannot produce high-quality noise. Instead, 2D Entropy is required for spatial correlations. Bethe Correction refines the KL estimate (improving low-resolution histograms), and Multilevel supervision further improves projection into the Gaussian distribution when combined with any configuration. Our loss applies only to the first three scales (1, 1/2, 1/4); the 1/8 scale is purely for visualization. Though results vary by input, this demonstrates that all components are essential. Further discussion on hyperparameters and learning rates in Appendix F.

3. Entropy-based Gaussian Regularization

In our quest to understand the mathematical underpinnings governing the *qualia* [39] of Gaussian samples, we focus on the local statistical character of a *typical* Gaussian realization. In diffusion models, this distinction is crucial: a latent variable may match a few global moments while still exhibiting strong spatial structure, making it clearly unlike a true Gaussian sample. Our regularizer is designed to measure this notion of Gaussianity from a single sample.

3.1. From Single Sample to Probability Distribution

With $\mathbf{x}_0 \in \mathbb{R}^d$ (*i.e.* an image from dataset) as an input sample and $G = \mathcal{N}(\mathbf{0}, \mathbf{I})$ a standard Gaussian distribution, a generic regularized objective for mapping a sample to its "closest" Gaussian counterpart can be written as

$$\mathbf{x}^\star = \arg\min_{\hat{\mathbf{x}}} \lambda_S \, \mathcal{D}_S(\mathbf{x}_0, \hat{\mathbf{x}}) + \lambda_G \, \mathcal{D}_G(\hat{\mathbf{x}}), \qquad (1)$$

where $\mathcal{D}_S$ measures how close the optimized solution should remain to the input sample $\mathbf{x}_0$, and $\mathcal{D}_G(\hat{\mathbf{x}})$ measures how compatible $\hat{\mathbf{x}}$ is with a Gaussian prior. The weighting terms λ_S and λ_G balance these two objectives.

A tempting choice for measuring the compatibility of a point with the Gaussian prior is the pointwise score $\mathcal{D}_G(\hat{\mathbf{x}}) = -\log G(\hat{\mathbf{x}})$, which—when combined with an ℓ_2 loss for $\mathcal{D}_S$—corresponds to a maximum a posteriori estimator (Appendix A). However, such a choice is fundamentally biased toward the *mode* of the Gaussian distribution, $\mathbf{0}$, rather than its high-dimensional *typical set*. In other words, it favors pointwise probability instead of the statistical structure of a genuine Gaussian sample. This suggests that Gaussianity should be formulated as a *distributional* property. A natural quantity that measures discrepancies between distributions is the Kullback–Leibler divergence

$$D_{\mathrm{KL}}(P \,\|\, Q) = \int_{\mathbf{x}\in\mathcal{X}} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} \, d\mathbf{x}, \qquad (2)$$

which satisfies $D_{\mathrm{KL}}(P \,\|\, Q) \geq 0$ and vanishes if and only if $p(\mathbf{x}) = q(\mathbf{x})$ almost everywhere. If $Q = G$, then $D_{\mathrm{KL}}(P \,\|\, G)$ is an ideal measure of Gaussianity.

We refer to computing Equation (2) for $Q = G$ as the *hard problem of Gaussianity*. This KL divergence is generally intractable in practice, since it requires evaluating all possible samples from the underlying data distribution P (or equivalently, the full expectation under P), while P itself is not necessarily known *a priori*. In practice, one can therefore only try to approximate it. One of our key ideas is to associate each optimized sample $\hat{\mathbf{x}}$ with an empirical

distribution $P_{\hat{\mathbf{x}}}$ induced by statistics extracted from $\hat{\mathbf{x}}$, and to define

$$\mathcal{D}_G(\hat{\mathbf{x}}) = D_{\mathrm{KL}}(P_{\hat{\mathbf{x}}} \,\|\, G). \tag{3}$$

Equation (3) captures the main conceptual shift of our method. We no longer ask whether a single point "belongs" to a Gaussian distribution. Instead, we ask whether the local statistics carried by that sample are consistent with those of a typical Gaussian realization.

3.2. An MRF-based Distributional Prior

The remaining question is how to formulate $P_{\hat{\mathbf{x}}}$ from a single sample. To circumvent this *hard problem of Gaussianity* in Eq. (2), previous regularizers typically either impose a heavily restricted family for $P_{\hat{\mathbf{x}}}$ or abandon distribution-matching entirely in favor of penalizing specific summary statistics. For instance, standard KL regularizers fall into the first category by implicitly assuming a fitted isotropic Gaussian, $P_{\hat{\mathbf{x}}} = \mathcal{N}(\boldsymbol{\mu}_{\hat{\mathbf{x}}}, \sigma^2_{\hat{\mathbf{x}}}\mathbf{I})$, which aligns only the first two global moments (see Appendix G). In the second category, methods bypass $P_{\hat{\mathbf{x}}}$ to constrain specific projections, such as the norm distribution [4, 13] or low-order spatial and spectral moments [21]. While these are meaningful approximations, they still match selected summaries rather than the sample's local joint statistics.

Our second key insight is that the distribution $P_{\hat{\mathbf{x}}}$ can be accurately approximated from a single sample by exploiting local statistical structures. Inspired by MRF formulations in classical computer vision [17, 31], we treat the optimized sample $\hat{\mathbf{x}}$ as an ergodic spatial process. This ergodicity, combined with a local neighborhood system and finite-order cliques, allows us to turn Eq. (2) into a computable objective. Under a standard pairwise-clique assumption, the density function $p_{\hat{\mathbf{x}}}(\mathbf{x})$ can then be factorized over the pixel lattice $\mathcal{V}$ as a product of unary (node) and pairwise (edge) potentials.

$$p_{\hat{\mathbf{x}}}(\mathbf{x}) = \frac{1}{Z} \prod_{i \in \mathcal{V}} \psi_i(x_i) \prod_{(i,j) \in \mathcal{E}} \psi_{ij}(x_i, x_j), \tag{4}$$

where $\mathcal{E}$ is the set of adjacent pixel pairs, ψ_i and ψ_{ij} are the potential functions (derived from the statistics of $\hat{\mathbf{x}}$), and Z is the partition function.

Simply using Equation (4) to define an objective remains intractable, since the partition function Z cannot, in general, be computed analytically. Fortunately, the same MRF structure that makes the problem meaningful also provides a principled approximation strategy through the *Bethe–Kikuchi cluster expansion* [5, 25, 65]. This method approximates the intractable global entropy (and thus the KL divergence) by a sum over the entropies of local clusters—in our case, unary and pairwise cliques—and corrects for their overlaps. Limiting this cluster expansion to nodes and edges (the standard Bethe approximation) yields our final objective function

$$\begin{aligned} D_{\mathrm{KL}}(P_{\hat{\mathbf{x}}} \,\|\, G) \;\approx\; & \underbrace{D_{\mathrm{KL}}\big(P_{\hat{\mathbf{x}},S^{(2)}} \,\|\, G_{S^{(2)}}\big)}_{\text{Spatial Entropy Term}} + \\ & \gamma\, \underbrace{D_{\mathrm{KL}}\big(P_{\hat{\mathbf{x}},S^{(1)}} \,\|\, G_{S^{(1)}}\big)}_{\text{Value Entropy Term}}, \end{aligned} \tag{5}$$

where γ is a correction term for over-counting, as prescribed by the Bethe approximation to account for nodes (pixels) that are part of multiple pairwise cliques. Detailed derivations can be found in Appendix B.

The two terms in Eq. (5) serve complementary purposes. The *Value Entropy Term* ($S^{(1)}$) measures the relative entropy between the 1D empirical distribution of pixel intensities, $P_{\hat{\mathbf{x}},S^{(1)}}(\mathbf{x})$, and the 1D target Gaussian $G_{S^{(1)}} = \mathcal{N}(0,1)$, thereby aligning the marginal value statistics of the sample with those of the Gaussian prior. The *Spatial Entropy Term* ($S^{(2)}$) measures the relative entropy between the 2D joint empirical distribution of adjacent pixel pairs, $P_{\hat{\mathbf{x}},S^{(2)}}(\mathbf{x})$, and the target 2D joint Gaussian $G_{S^{(2)}} = \mathcal{N}(0, I_2)$. This term is central to capturing typicality, as it penalizes local statistical dependencies, which are absent in an ideal Gaussian field. Figure 1 illustrates these two quantities on representative examples. In practice, minimizing the objective in Eq. (5) requires two further components: a differentiable formulation of the relative entropy terms and an efficient optimization strategy.

3.3. Differentiable Relative Entropy Estimation

At the core of our objective function in Eq. (5) is the sum of the divergences $D_{KL}(P_{\hat{\mathbf{x}}} \,\|\, G)$ for both the unary ($S^{(1)}$) and pairwise ($S^{(2)}$) empirical distributions. We compute each divergence by separating it into its two components:

$$D_{KL}(P_{\hat{\mathbf{x}}} \,\|\, G) = \underbrace{H(P_{\hat{\mathbf{x}}}, G)}_{\text{Cross-Entropy}} - \underbrace{H(P_{\hat{\mathbf{x}}})}_{\text{Differential Entropy}}. \tag{6}$$

For either empirical distribution, we denote by $\mathcal{V} = \{\mathbf{v}_i\}_{i=1}^N$ the corresponding set of N d-dimensional vectors extracted from the sample $\hat{\mathbf{x}}$: for $S^{(1)}$, $\mathbf{v}_i \in \mathbb{R}$ corresponds to a pixel value ($d = 1$), whereas for $S^{(2)}$, $\mathbf{v}_i \in \mathbb{R}^2$ corresponds to an immediate neighboring pair ($d = 2$). The same derivation below applies to both cases.

Cross-Entropy $H(P_{\hat{\mathbf{x}}}, G) = -\mathbb{E}_{\mathbf{v} \sim P_{\hat{\mathbf{x}}}}[\log G(\mathbf{v})]$**.** Since our target distribution $G = \mathcal{N}(0, I_d)$ is analytic, its log-probability is a simple, differentiable function:

$$\log G(\mathbf{v}) = -\frac{d}{2}\log(2\pi) - \frac{1}{2}\|\mathbf{v}\|_2^2. \tag{7}$$

We estimate the cross-entropy using a standard Monte Carlo approximation by averaging over the N elements of a sam-

ple as:

$$H(P_{\hat{\mathbf{x}}}, G) \approx -\frac{1}{N}\sum_{i=1}^{N} \log G(\mathbf{v}_i). \tag{8}$$

This term is fully differentiable with respect to the components of each $\mathbf{v}_i$.

Differential Entropy $H(P_{\hat{\mathbf{x}}}) = -\mathbb{E}_{\mathbf{v}\sim P_{\hat{\mathbf{x}}}}[\log P_{\hat{\mathbf{x}}}(\mathbf{v})]$ is more complex, as $P_{\hat{\mathbf{x}}}$ is only defined by the set of samples $\mathcal{V}$. To create a differentiable estimate, we first approximate the continuous density $P_{\hat{\mathbf{x}}}(\mathbf{v})$ using Kernel Density Estimation (KDE) with a Gaussian kernel $\mathcal{K}_\sigma$:

$$P_{\hat{\mathbf{x}}}(\mathbf{v}) \approx \hat{p}(\mathbf{v}) = \frac{1}{N}\sum_{j=1}^{N} \mathcal{K}_\sigma(\mathbf{v} - \mathbf{v}_j), \tag{9}$$

where σ is the kernel bandwidth, a hyperparameter often set using a heuristic like Scott's rule ($\sigma = N^{-1/(d+4)}$) [54].

With this differentiable density estimator, we can compute the differential entropy using a Monte Carlo estimate. The log-density $\log \hat{p}(\mathbf{v}_i)$ is evaluated at each sample $\mathbf{v}_i$ from our set and averaged as

$$\begin{aligned} H(P_{\hat{\mathbf{x}}}) &\approx -\frac{1}{N}\sum_{i=1}^{N} \log\left(\hat{p}(\mathbf{v}_i) + \epsilon\right) \\ &= -\frac{1}{N}\sum_{i=1}^{N} \log\left(\frac{1}{N}\sum_{j=1}^{N} \mathcal{K}_\sigma(\mathbf{v}_i - \mathbf{v}_j) + \epsilon\right), \end{aligned} \tag{10}$$

where ϵ is a small constant for numerical stability. The complete expression for D_{KL} is the difference between Eq. 8 and Eq. 10. This entire formulation is end-to-end differentiable with respect to the original pixel values in $\hat{\mathbf{x}}$, which constitute the vectors $\mathbf{v}_i$.

Naively, this formulation requires a pairwise $O(N^2)$ computation over the N sample vectors (pixels or pixel pairs). However, we can achieve linear complexity without noticeable quality loss by computing pairwise distances against a fixed set of bins (e.g. 128), which we use in Table 1. Since this cost is incurred per-sample during optimization, not during large-scale model training, it remains a viable solution for our target applications. More discussion regarding efficiency can be found in Appendix E.

3.4. Multi-Scale Optimization

Our MRF assumption captures only local *pairwise* dependencies between neighboring pixels in the sample. Consequently, relationships between pixels that are farther apart are not explicitly constrained by the objective, and long-range correlations may still persist in $\hat{\mathbf{x}}$ even when local statistics are well matched to those of G. To address this limitation and encourage statistical independence across larger spatial scales, we employ a multi-level optimization scheme. Rather than applying the loss in Equation (5) only at the original resolution, we also evaluate it on progressively downsampled versions of $\hat{\mathbf{x}}$.

We construct a pyramid of samples $\{\hat{\mathbf{x}}_k\}_{k=0}^{L-1}$, where $\hat{\mathbf{x}}_0 = \hat{\mathbf{x}}$ denotes the full-resolution sample and each $\hat{\mathbf{x}}_k$ for $k > 0$ is obtained by downsampling the previous level. Specifically, we apply mean pooling over 2×2 blocks and then rescale the result by $\sqrt{n}$, where n is the number of aggregated pixels, in order to preserve variance. For 2×2 pooling, this corresponds to $n = 4$ and thus a rescaling factor of $\sqrt{4} = 2$. Under this variance-preserving transformation, the target distribution at every level remains the standard normal $G = \mathcal{N}(0, I)$. The KL-based objective of Equation (6) can be evaluated at each scale, which yields the final objective

$$\mathcal{L}_{\text{full}}(\hat{\mathbf{x}}) = \sum_{k=0}^{L-1} \alpha_k D_{KL}\left(P_{\hat{\mathbf{x}}_k} \,\|\, G\right), \tag{11}$$

where α_k are weights that balance the KL objective across different scales. This multi-level formulation penalizes spatial correlations over progressively larger neighborhoods, making the final sample $\hat{\mathbf{x}}$ a more faithful representative of the Gaussian typical set. In practice, we use three levels ($L = 3$) for all examples in this paper. Figure 2 illustrates these statistics across the different resolution levels.

KL	Pix2Pix-Zero	ReNO	ReNoise	Hwang et al.	**Ours**
0.4063 ±0.0227	2.6374 ±0.1207	0.4553 ±0.0025	3.0938 ±0.2804	0.7148 ±0.0075	**15.0191 ±0.1053**

Table 1. **Time/step (ms)** for various baselines and our approach.

4. Experiments & Validation

4.1. Ablation Studies

To assess the contribution of each component, a cumulative ablation study is performed (Figure 2). A structured checkerboard pattern is used as the input, providing a clear visual baseline that isolates the effect of each regularization term. The formulation is then built progressively by adding the 1D entropy term, the 2D entropy term, and the Bethe correction. At each stage, results are shown both with and without the multilevel optimization. Although the impact of each term depends on the input and hyperparameter choice (more in Appendix F), this example clearly illustrates how the full formulation comes together:

- **1D (Value) Entropy** successfully tames the 1D marginal histogram (middle rows), matching all first-order statistics. However, it fails to address the input's strong spatial correlations.
- **2D (Spatial) Entropy** begins to break down these local correlations when added at the full scale.

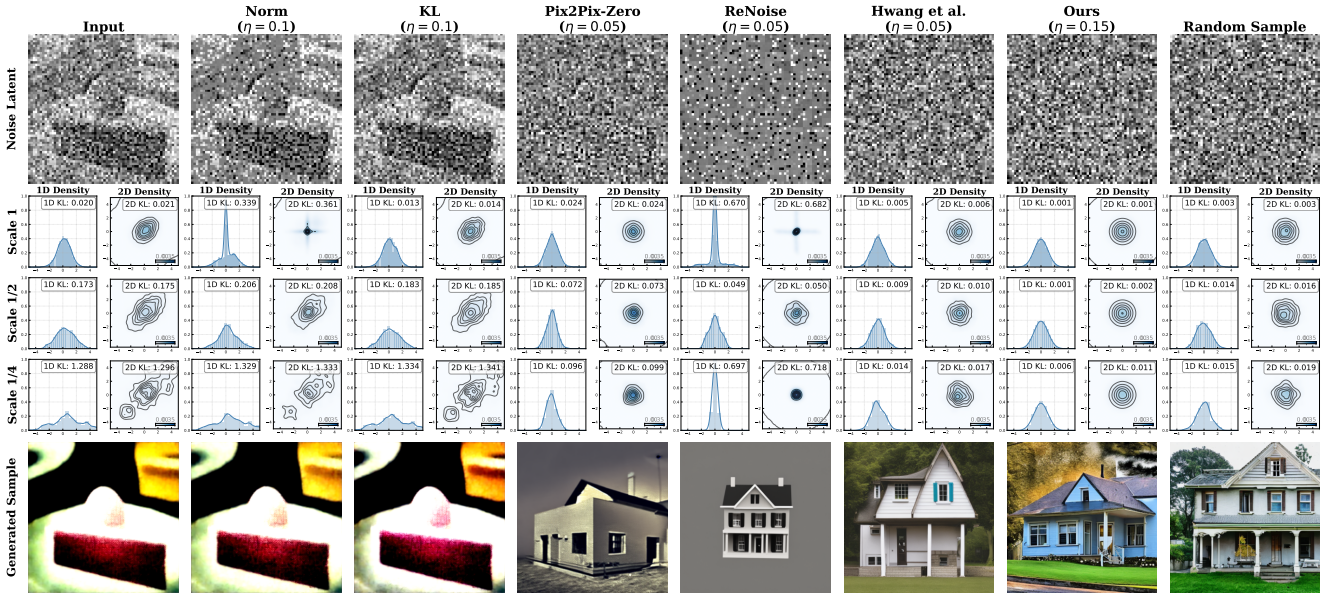

Figure 3. **Baseline Comparisons.** We evaluate baseline methods on optimizing an input latent towards Gaussian noise. The input is a cake image latent mixed with 50% noise to reflect typical use cases containing both noise and hidden structure. Rows display the input/optimized latent, multiscale 1D/2D densities, and the final generated image (*A photo of a house*). To ensure we isolate the loss function's projection capability from optimizer-induced stochasticity, we individually tuned the learning rate (η) for each baseline to a value that both stabilizes the noise and maximizes output quality. All methods were optimized for 5000 steps to ensure convergence. Our method yields a more accurate Gaussian noise sample than the baselines, achieving generated image quality similar to perfectly random noise.

- **Bethe Correction** further refines the KL estimate, and we empirically observe that it improves statistical convergence at lower-resolution scales.
- **Multilevel Scheme**, when applied alongside any of these configurations, consistently penalizes long-range correlations, forcing the sample to become statistically and visually closer to true Gaussian noise.

4.2. Comparisons with Previous Work

After evaluating the contribution of each component, we benchmark the complete regularizer against several existing and concurrent methods. In Figure 3, a noisy image latent more representative of practical use cases is used to visualize the projection produced by each baseline. The comparisons include **KL Divergence Loss** [27], which directly minimizes the KL divergence between sample statistics and the Gaussian distribution; **Norm Regularization (ReNO)** [13, 50], based on the negative log-likelihood of the vector norm; **Pix2Pix-Zero** [44], which combines pairwise autocorrelation and KL losses; **ReNoise** [15], which combines autocorrelation with a patch-based KL divergence; and **Hwang et al.** [21], a concurrent method that regularizes the first and second moments together with the power spectrum. Figure 3 shows that our full method produces samples that are statistically and visually more consistent with random Gaussian noise, yielding better images.

5. Applications

We demonstrate the effectiveness of our typicality-preserving objective $\mathcal{L}_{\text{full}}$ on several downstream tasks that rely on constrained optimization in the latent space of diffusion models. A primary application is **reward-guided generation**, where the goal is to find a noise latent $\mathbf{x}^*$ that maximizes a given reward R (e.g., aesthetic score, brightness, prompt-alignment) while still remaining in the Gaussian typical set:

$$\mathbf{x}^* = \arg\min_{\hat{\mathbf{x}}} -R(\hat{\mathbf{x}}) + \lambda \mathcal{L}_{\text{full}}(\hat{\mathbf{x}}) \tag{12}$$

Without a robust regularizer, this optimization can easily hack the reward under high learning rates, drifting away from the valid noise manifold and producing non-image artifacts. Our loss $\mathcal{L}_{\text{full}}(\hat{\mathbf{x}})$ effectively prevents this overfitting, as shown in the following applications.

5.1. Aesthetic Image Generation

We optimize the initial noise latent to maximize a pretrained Aesthetic Score [53]. As shown both qualitatively and quantitatively in Figure 4 and Table 2, our regularizer (*Ours*) constrains the optimization to produce images that are more aesthetic and natural. It avoids the over-saturated colors and unnatural textures produced by unregularized optimization (*No Reg.*) or weaker regularizers. More details can be found in the Appendix D.

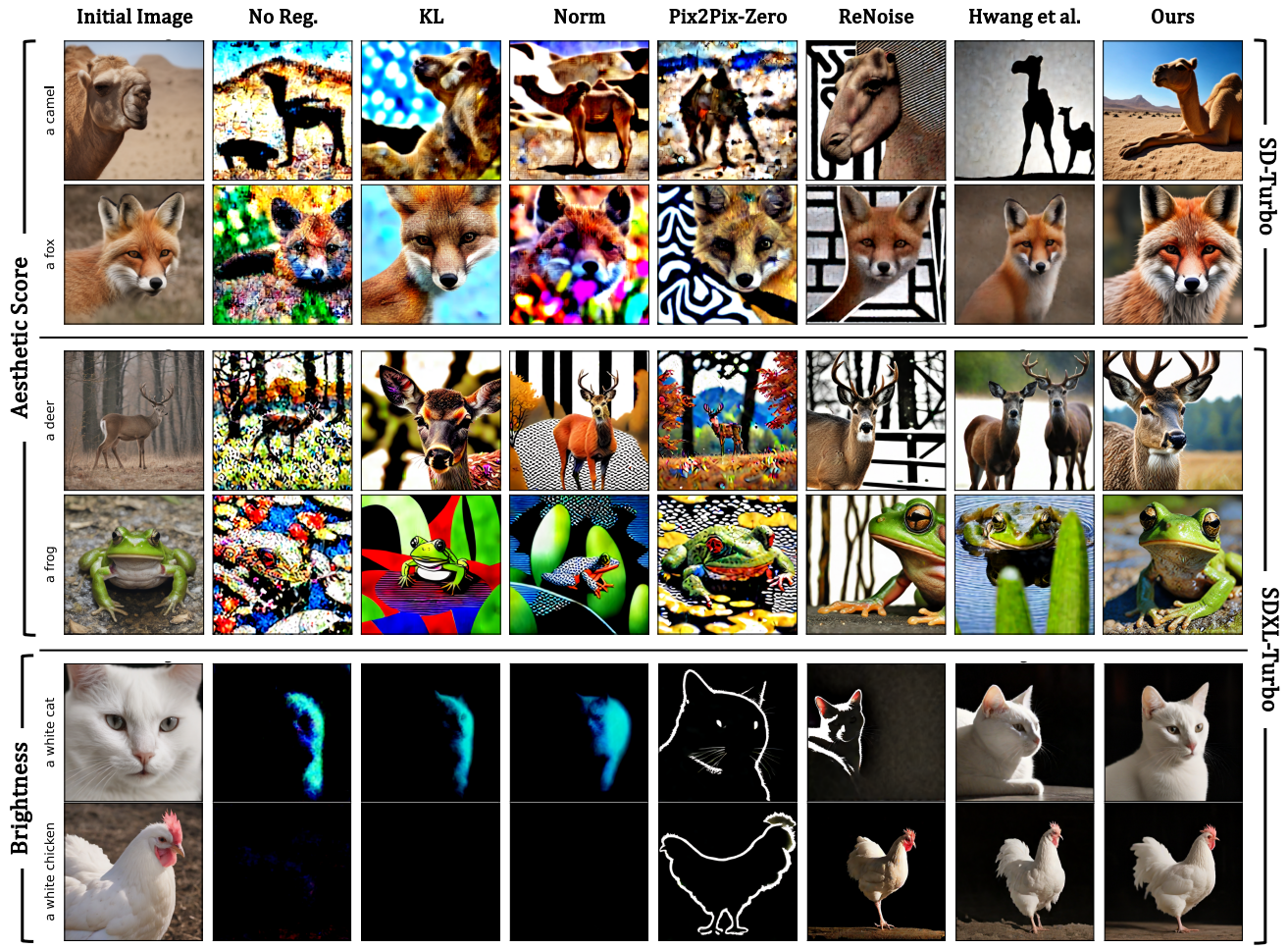

Figure 4. **Reward Alignment Image Generation.** We show some qualitative comparisons between various regularization techniques on two reward alignment tasks: aesthetic score optimization (top), brightness minimization (bottom). By better preserving the Gaussian distribution, our optimized noise generates artifact-free images with neither over-saturated colors nor undesirable splotches.

5.2. Attribute Control

Our method can also control simple image attributes. We evaluate this capability on brightness minimization [57]. The bottom rows of Figure 4 and the results in Table 2 show, both qualitatively and quantitatively, that our regularizer successfully steers generation toward the desired attribute. Unlike the unregularized baseline (*No Reg.*), it does so without introducing severe artifacts, effectively keeping the latent $\hat{\mathbf{x}}$ within the space of image-producing noise.

5.3. Model-free Image-to-Noise Matching

Finally, we explore what is perhaps the most surprising consequence of optimizing Equation (1): image variant generation/inversion *without* access to the diffusion model itself. Earlier works have noted that diffusion noise and generated images can be strongly correlated in structure [24, 32, 45]. Here we further extend this intuition: given only a target

Method	Aesth. ↑	CLIP ↑	HPSv2 ↑	ImgRwd ↑	PickSc. ↑
Initial	5.440	24.997	0.292	0.822	22.543
No Reg.	**7.975**	19.682	0.209	-0.950	18.595
KL [27]	6.594	22.235	0.255	0.272	19.938
ReNO [13]	6.626	21.796	0.250	0.188	19.806
Pix2Pix-Zero [44]	7.031	20.967	0.244	0.071	19.454
ReNoise [15]	5.806	23.286	0.257	0.248	20.385
Hwang et al. [21]	5.863	**25.019**	0.277	0.729	21.381
Ours	6.478	24.574	**0.288**	**0.826**	**21.625**

Method	Aesth. ↑	Bright ↓	CLIP ↑	HPSv2 ↑	ImgRwd ↑	PickSc. ↑
Initial	5.418	0.504	28.039	0.306	1.179	23.197
No Reg.	3.662	0.021	18.606	0.110	-2.275	17.569
KL [27]	4.064	**0.005**	19.360	0.101	-2.232	18.096
ReNO [13]	4.077	0.008	20.370	0.110	-2.119	18.198
Pix2Pix-Zero [44]	4.717	0.070	23.041	0.178	-1.579	19.020
ReNoise [15]	5.339	0.108	27.080	0.240	-0.103	20.788
Hwang et al. [21]	5.655	0.228	**28.169**	0.287	0.925	22.467
Ours	**5.768**	0.270	27.914	**0.298**	**1.038**	**22.823**

Table 2. **Aesthetic Image Generation (top) and Brightness Minimization Reward (bottom) with SDXL-Turbo [52].** We use a set of animal prompts from DDPO [6] in the format *A photo of [animal]* (top) and *A photo of a white [animal]* (bottom).

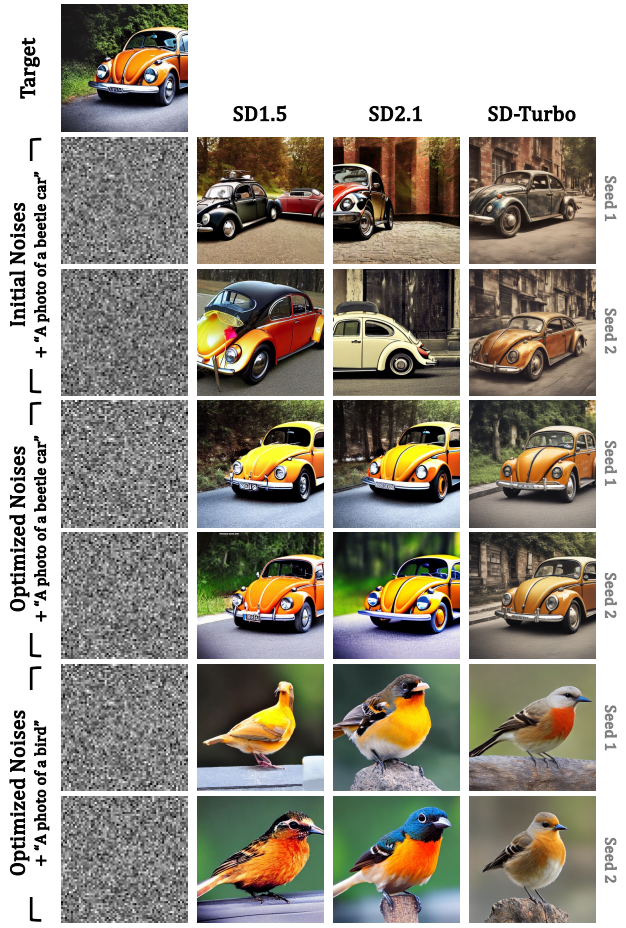

Figure 5. **Model-free Image-to-Noise Matching**. We use Pearson correlation with a target image as a reward. Our regularization enables the creation of noise samples without using diffusion models that produce images with similar layout and structure when given corresponding prompts. At the same time, these noise samples can still generate arbitrary images under different prompts. Notably, the resulting noises work across different models.

image, our method can recover a corresponding Gaussian noise latent *without ever querying the denoiser*, making the process entirely model-free and model-agnostic. To the best of our knowledge, this is the first demonstration of such an effect. Concretely, we optimize the initial noise to maximize Pearson correlation with a target image as a reward, while our regularizer keeps the solution within the Gaussian typical set. This reward can also be seen as the $\mathcal{D}_S$ term in Equation (1). As shown in Figure 5, starting from two random initializations yields optimized latents whose generated images exhibit strikingly similar layout and structure to the target. Although the regularization weight λ must be tuned per sample, the phenomenon is consistent across multiple diffusion models, including SD1.5, SD2.1 [48], and SD-Turbo [52]. Importantly, the recovered latent is not merely overfitted to the source image: when paired with a different prompt, such as *a photo of a bird*, it still generates a high-quality matching image, indicating that it remains a general-purpose Gaussian latent rather than a degenerate encoding. While the inherent spatial correlation between initial noise and generated images explains this structural alignment, refining this model-free recovery into a mathematically faithful, high-fidelity inversion technique remains an exciting open challenge.

6. Discussion & Conclusion

"It is like being mistaken for a mode, when one was always meant to be a distribution," answered the noise, during a short break after denoising step 25. In this work, we took that perspective seriously by casting Gaussian regularization for arbitrary samples as a distributional matching problem. Our solution lifts each sample into an empirical distribution, modeled as a pairwise Markov Random Field, and measures Gaussianity through the KL divergence $D_{\mathrm{KL}}(P_{\hat{\mathbf{x}}} \,\|\, G)$. Using a Bethe–Kikuchi approximation, we derive a tractable objective that combines 1D marginal entropy, 2D spatial entropy, and a multi-scale strategy to remove longer-range correlations. The resulting regularizer is robust, differentiable, and deterministic, and experiments demonstrate its effectiveness in stabilizing latent optimization for reward-guided generation and model-free image-to-noise matching while preventing reward hacking.

Our method still has limitations. The reliance on Kernel Density Estimation (KDE) to compute the differential entropy, though providing a smooth estimate, is computationally expensive. Furthermore, while our multi-scale MRF is a significant improvement over simpler priors, its underlying pairwise assumption is still a local approximation and may not perfectly destroy all global, long-range dependencies (see Fig. 6). Future work could explore more advanced and efficient implicit distributions beyond pairwise MRFs, as well as alternatives to KDE to improve performance.

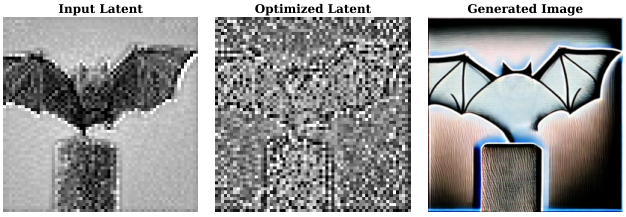

Figure 6. **Failure case.** As shown on the left, when optimizing from a purely clean latent (*A photo of a bat*), our multilevel MRF assumption may still fail to resolve complex, long-range structures (middle). This yields the degraded diffusion outputs seen on the right (*A photo of a house*). In practical applications, however, inputs are rarely this far removed from the target noise distribution.

References

[1] Donghoon Ahn, Jiwon Kang, Sanghyun Lee, Jaewon Min, Minjae Kim, Wooseok Jang, Hyoungwon Cho, Sayak Paul, SeonHwa Kim, Eunju Cha, Kyong Hwan Jin, and Seungryong Kim. A Noise is Worth Diffusion Guidance, 2024. arXiv:2412.03895 [cs]. 2

[2] Kolmogorov An. Sulla determinazione empirica di una legge didistribuzione. *Giorn Dell'inst Ital Degli Att*, 4:89–91, 1933. 2

[3] T. W. Anderson and D. A. Darling. Asymptotic Theory of Certain "Goodness of Fit" Criteria Based on Stochastic Processes. *The Annals of Mathematical Statistics*, 23(2):193–212, 1952. 2

[4] Heli Ben-Hamu, Omri Puny, Itai Gat, Brian Karrer, Uriel Singer, and Yaron Lipman. D-Flow: Differentiating through Flows for Controlled Generation, 2024. arXiv:2402.14017 [cs]. 1, 2, 4

[5] H. A. Bethe. Statistical Theory of Superlattices. In *Selected Works of Hans A Bethe*, pages 245–270. WORLD SCIENTIFIC, 1997. 4

[6] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *The Twelfth International Conference on Learning Representations*. 7, 4

[7] Ryan Burgert, Yuancheng Xu, Wenqi Xian, Oliver Pilarski, Pascal Clausen, Mingming He, Li Ma, Yitong Deng, Lingxiao Li, Mohsen Mousavi, Michael Ryoo, Paul Debevec, and Ning Yu. Go-with-the-Flow: Motion-Controllable Video Diffusion Models Using Real-Time Warped Noise, 2025. arXiv:2501.08331 [cs]. 2

[8] Pascal Chang, Jingwei Tang, Markus Gross, and Vinicius C. Azevedo. How I Warped Your Noise: a Temporally-Correlated Noise Prior for Diffusion Models, 2025. arXiv:2504.03072 [cs]. 2

[9] Changgu Chen, Libing Yang, Xiaoyan Yang, Lianggangxu Chen, Gaoqi He, CHangbo Wang, and Yang Li. FIND: Fine-tuning Initial Noise Distribution with Policy Optimization for Diffusion Models, 2024. arXiv:2407.19453 [cs]. 2

[10] Scott Chen and Ramesh Gopinath. Gaussianization. In *Advances in Neural Information Processing Systems*. MIT Press, 2000. 2

[11] Omer Dahary, Yehonathan Cohen, Or Patashnik, Kfir Aberman, and Daniel Cohen-Or. Be Decisive: Noise-Induced Layouts for Multi-Subject Generation, 2025. arXiv:2505.21488 [cs]. 2

[12] Giannis Daras, Weili Nie, Karsten Kreis, Alex Dimakis, Morteza Mardani, Nikola Borislavov Kovachki, and Arash Vahdat. Warped Diffusion: Solving Video Inverse Problems with Image Diffusion Models, 2024. arXiv:2410.16152 [cs]. 2

[13] Luca Eyring, Shyamgopal Karthik, Karsten Roth, Alexey Dosovitskiy, and Zeynep Akata. ReNO: Enhancing One-step Text-to-Image Models through Reward-based Noise Optimization, 2024. arXiv:2406.04312 [cs]. 1, 2, 4, 6, 7, 8

[14] Luca Eyring, Shyamgopal Karthik, Alexey Dosovitskiy, Nataniel Ruiz, and Zeynep Akata. Noise Hypernetworks: Amortizing Test-Time Compute in Diffusion Models, 2025. arXiv:2508.09968 [cs]. 2

[15] Daniel Garibi, Or Patashnik, Andrey Voynov, Hadar Averbuch-Elor, and Daniel Cohen-Or. Renoise: Real image inversion through iterative noising. In *European Conference on Computer Vision*, pages 395–413. Springer, 2024. 6, 7, 3, 4, 8

[16] Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. Preserve Your Own Correlation: A Noise Prior for Video Diffusion Models, 2024. arXiv:2305.10474 [cs]. 2

[17] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984. 4

[18] Or Greenberg, Eran Kishon, and Dani Lischinski. S2ST: Image-to-Image Translation in the Seed Space of Latent Diffusion, 2023. arXiv:2312.00116 [cs]. 2

[19] Xiefan Guo, Jinlin Liu, Miaomiao Cui, Jiankai Li, Hongyu Yang, and Di Huang. InitNO: Boosting Text-to-Image Diffusion Models via Initial Noise Optimization, 2024. arXiv:2404.04650 [cs]. 1, 2

[20] Xingchang Huang, Corentin Salaün, Cristina Vasconcelos, Christian Theobalt, Cengiz Öztireli, and Gurprit Singh. Blue noise for diffusion models, 2024. arXiv:2402.04930 [cs]. 2

[21] Jisung Hwang, Jaihoon Kim, and Minhyuk Sung. Moment- and Power-Spectrum-Based Gaussianity Regularization for Text-to-Image Models, 2025. arXiv:2509.07027 [cs]. 1, 2, 4, 6, 7, 3, 8

[22] Carlos M. Jarque and Anil K. Bera. Efficient tests for normality, homoscedasticity and serial independence of regression residuals. *Economics Letters*, 6(3):255–259, 1980. 2

[23] Korrawe Karunratanakul, Konpat Preechakul, Emre Aksan, Thabo Beeler, Supasorn Suwajanakorn, and Siyu Tang. Optimizing Diffusion Noise Can Serve As Universal Motion Priors, 2024. arXiv:2312.11994 [cs]. 2

[24] Valentin Khrulkov, Gleb Ryzhakov, Andrei Chertkov, and Ivan Oseledets. Understanding DDPM latent codes through optimal transport. In *The Eleventh International Conference on Learning Representations*, 2023. 7

[25] Ryoichi Kikuchi. A Theory of Cooperative Phenomena. *Physical Review*, 81(6):988–1003, 1951. Publisher: American Physical Society. 4

[26] Kwanyoung Kim and Sanghyun Kim. Model Already Knows the Best Noise: Bayesian Active Noise Selection via Attention in Video Diffusion Model, 2025. arXiv:2505.17561 [cs]. 2

[27] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1, 2, 6, 7, 4, 8

[28] Juil Koo, Paul Guerrero, Chun-Hao Paul Huang, Duygu Ceylan, and Minhyuk Sung. VideoHandles: Editing 3D Object Compositions in Videos Using Video Generative Priors, 2025. arXiv:2503.01107 [cs]. 2

[29] Min-Seop Kwak, Donghoon Ahn, Ines Hyeonsu Kim, Jin-Hwa Kim, and Seungryong Kim. Geometry-Aware Score

Distillation via 3D Consistent Noising and Gradient Consistency Modeling, 2024. arXiv:2406.16695 [cs]. 2

[30] Shuangqi Li, Hieu Le, Jingyi Xu, and Mathieu Salzmann. All Seeds Are Not Equal: Enhancing Compositional Text-to-Image Generation with Reliable Random Seeds, 2025. arXiv:2411.18810 [cs]. 2

[31] Stan Z Li. *Markov random field modeling in computer vision*. Springer Science & Business Media, 2012. 4

[32] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *11th International Conference on Learning Representations, ICLR 2023*, 2023. 7

[33] Chao Liu and Arash Vahdat. EquiVDM: Equivariant Video Diffusion Models with Temporally Consistent Noise, 2025. arXiv:2504.09789 [cs]. 2

[34] Yongzhe Lyu, Yu Wu, Yutian Lin, and Bo Du. IS-Diff: Improving Diffusion-Based Inpainting with Better Initial Seed, 2025. arXiv:2509.11638 [cs]. 2

[35] Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, and Saining Xie. Inference-Time Scaling for Diffusion Models beyond Scaling Denoising Steps, 2025. arXiv:2501.09732 [cs]. 2

[36] Jiafeng Mao, Xueting Wang, and Kiyoharu Aizawa. Guided Image Synthesis via Initial Image Editing in Diffusion Model. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 5321–5329, 2023. arXiv:2305.03382 [cs]. 2

[37] Jiafeng Mao, Xueting Wang, and Kiyoharu Aizawa. The Lottery Ticket Hypothesis in Denoising: Towards Semantic-Driven Initialization, 2024. arXiv:2312.08872 [cs]. 2

[38] Boming Miao, Chunxiao Li, Xiaoxiao Wang, Andi Zhang, Rui Sun, Zizhe Wang, and Yao Zhu. Noise Diffusion for Enhancing Semantic Faithfulness in Text-to-Image Synthesis, 2024. arXiv:2411.16503 [cs]. 1, 2

[39] Thomas Nagel. What is it like to be a bat? *The Philosophical Review*, 83(4):435–450, 1974. 3

[40] Zachary Novack, Julian McAuley, Taylor Berg-Kirkpatrick, and Nicholas J. Bryan. DITTO: Diffusion Inference-Time T-Optimization for Music Generation, 2024. arXiv:2401.12179 [cs]. 2

[41] H. Nyquist. Thermal Agitation of Electric Charge in Conductors. *Physical Review*, 32(1):110–113, 1928. 1

[42] Krzysztof Ostrowski, Michał Piasecki, Małgorzata Kudelska, Patryk Bartkowiak, Haohong Wang, and Brian Bullock. Enhancing Video Stylization with Integral Noise. In *2025 International Conference on Computing, Networking and Communications (ICNC)*, pages 622–628, 2025. ISSN: 2473-7585. 2

[43] Athanasios Papoulis and S. Unnikrishna Pillai. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 2002. Google-Books-ID: YYwQAQAAIAAJ. 1

[44] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot Image-to-Image Translation, 2023. arXiv:2302.03027 [cs]. 1, 6, 7, 3, 4, 8

[45] Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky T Chen. Multisample flow matching: Straightening flows with minibatch couplings. *ICML 2023*, 2023. 7

[46] Zipeng Qi, Lichen Bai, Haoyi Xiong, and Zeke Xie. Not All Noises Are Created Equally:Diffusion Noise Selection and Optimization, 2024. arXiv:2407.14041 [cs]. 2

[47] Haonan Qiu, Menghan Xia, Yong Zhang, Yingqing He, Xintao Wang, Ying Shan, and Ziwei Liu. FreeNoise: Tuning-Free Longer Video Diffusion via Noise Rescheduling, 2024. arXiv:2310.15169 [cs]. 2

[48] 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*, pages 10684–10695, 2022. 8

[49] Roey Ron, Guy Tevet, Haim Sawdayee, and Amit H. Bermano. HOIDiNi: Human-Object Interaction through Diffusion Noise Optimization, 2025. arXiv:2506.15625 [cs]. 2

[50] Dvir Samuel, Rami Ben-Ari, Nir Darshan, Haggai Maron, and Gal Chechik. Norm-guided latent space exploration for text-to-image generation, 2023. arXiv:2306.08687 [cs]. 2, 6

[51] Dvir Samuel, Rami Ben-Ari, Simon Raviv, Nir Darshan, and Gal Chechik. Generating images of rare concepts using pre-trained diffusion models, 2023. arXiv:2304.14530 [cs]. 2

[52] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pages 87–103. Springer, 2024. 7, 8

[53] C. Schuhmann. Laion aesthetics. https://laion.ai/blog/laion-aesthetics, 2022. 6

[54] David W Scott. On optimal and data-based histograms. *Biometrika*, 66(3):605–610, 1979. 5

[55] S Shaphiro and MBJB Wilk. An analysis of variance test for normality. *Biometrika*, 52(3):591–611, 1965. 2

[56] Moran Shkolnik, Brian Chmiel, Ron Banner, Gil Shomron, Yury Nahshan, Alex Bronstein, and Uri Weiser. Robust Quantization: One Model to Rule Them All, 2020. arXiv:2002.07686 [cs]. 2

[57] Zhiwei Tang, Jiangweizhi Peng, Jiasheng Tang, Mingyi Hong, Fan Wang, and Tsung-Hui Chang. Inference-Time Alignment of Diffusion Models with Direct Noise Optimization, 2024. arXiv:2405.18881 [cs]. 1, 2, 7

[58] Jente Vandersanden, Sascha Holl, Xingchang Huang, and Gurprit Singh. Edge-preserving noise for diffusion models, 2025. arXiv:2410.01540 [cs]. 2

[59] Bram Wallace, Akash Gokul, Stefano Ermon, and Nikhil Naik. End-to-End Diffusion Latent Optimization Improves Classifier Guidance, 2023. arXiv:2303.13703 [cs]. 2

[60] Ruoyu Wang, Huayang Huang, Ye Zhu, Olga Russakovsky, and Yu Wu. The Silent Assistant: NoiseQuery as Implicit Guidance for Goal-Driven Image Generation, 2025. arXiv:2412.05101 [cs]. 2

[61] Xingrui Wang, Xin Li, and Zhibo Chen. CoNo: Consistency Noise Injection for Tuning-free Long Video Diffusion, 2024. arXiv:2406.05082 [cs]. 2

[62] Yanghao Wang and Long Chen. Noise Matters: Optimizing Matching Noise for Diffusion Classifiers, 2025. arXiv:2508.11330 [cs]. 2

[63] Katherine Xu, Lingzhi Zhang, and Jianbo Shi. Good Seed Makes a Good Crop: Discovering Secret Seeds in Text-to-Image Diffusion Models, 2025. arXiv:2405.14828 [cs]. 2

[64] Runjie Yan, Yinbo Chen, and Xiaolong Wang. Consistent Flow Distillation for Text-to-3D Generation, 2025. arXiv:2501.05445 [cs]. 2

[65] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. In *Exploring artificial intelligence in the new millennium*, pages 239–269. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003. 4

[66] Sunjae Yoon, Gwanhyeong Koo, Ji Woo Hong, and Chang D. Yoo. DNI: Dilutional Noise Initialization for Diffusion Video Editing, 2024. arXiv:2409.13037 [cs]. 2

[67] Taehoon Yoon, Yunhong Min, Kyeongmin Yeo, and Minhyuk Sung. Psi-Sampler: Initial Particle Sampling for SMC-Based Inference-Time Reward Alignment in Score Models, 2025. arXiv:2506.01320 [cs]. 2

[68] Yunlong Yuan, Yuanfan Guo, Chunwei Wang, Wei Zhang, Hang Xu, and Li Zhang. FreqPrior: Improving Video Diffusion Models with Frequency Filtering Gaussian Noise, 2025. arXiv:2502.03496 [eess]. 2

[69] Kevin Zhai, Utsav Singh, Anirudh Thatipelli, Souradip Chakraborty, Anit Kumar Sahu, Furong Huang, Amrit Singh Bedi, and Mubarak Shah. MIRA: Towards Mitigating Reward Hacking in Inference-Time Alignment of T2I Diffusion Models, 2025. arXiv:2510.01549 [cs]. 1, 2

[70] Kaifeng Zhao, Gen Li, and Siyu Tang. DartControl: A Diffusion-Based Autoregressive Motion Model for Real-Time Text-Driven Motion Control, 2025. arXiv:2410.05260 [cs]. 2

[71] Zikai Zhou, Shitong Shao, Lichen Bai, Shufei Zhang, Zhiqiang Xu, Bo Han, and Zeke Xie. Golden Noise for Diffusion Models: A Learning Framework, 2025. arXiv:2411.09502 [cs]. 2

What Is It Like to Be a Noise?
An Entropy-based Gaussian Noise Regularization for Diffusion Models

Supplementary Material

A. Soft Projection as a MAP Estimate

In this section, we provide additional details on the connection between geometric projection and the Maximum A Posteriori estimate, as mentioned in Section 3.1.

A.1. From Hard Projection to Soft Projection

One could think of regularization in our model as projecting an input sample $\mathbf{x}_0 \in \mathbb{R}^d$ onto a target probability distribution $p(\mathbf{y})$. As mentioned, this can be framed as a *soft* generalization of classical geometric projection.

A standard geometric projection of $\mathbf{x}_0$ onto a convex set $\mathcal{C}$ is formulated as a constrained optimization:

$$\Pi_{\mathcal{C}}(\mathbf{x}_0) = \arg\min_{\mathbf{y}\in\mathcal{C}} \frac{1}{2}\|\mathbf{x}_0 - \mathbf{y}\|_2^2. \tag{13}$$

Using an indicator function $\mathbb{I}_{\mathcal{C}}(\mathbf{y})$ ($\mathbb{I}_{\mathcal{C}}(\mathbf{y}) = 0$ if $\mathbf{y} \in \mathcal{C}$ and $\mathbb{I}_{\mathcal{C}}(\mathbf{y}) \to \infty$ otherwise), the above formulation becomes an unconstrained problem:

$$\Pi_{\mathcal{C}}(\mathbf{x}_0) = \arg\min_{\mathbf{y}\in\mathbb{R}^d} \left(\frac{1}{2}\|\mathbf{x}_0 - \mathbf{y}\|_2^2 + \mathbb{I}_{\mathcal{C}}(\mathbf{y})\right). \tag{14}$$

To generalize this concept, we consider our constraint to be the distribution $p(\mathbf{y})$ itself. We replace the *hard* binary penalty $\mathbb{I}_{\mathcal{C}}(\mathbf{y})$ with a *soft* one, specifically the negative log-probability $R(\mathbf{y}) = -\log p(\mathbf{y})$. This term penalizes points $\mathbf{y}$ that are highly out-of-distribution (where $p(\mathbf{y}) \to 0$). Our soft projection formulation is:

$$\mathbf{y}^* = \arg\min_{\mathbf{y}\in\mathbb{R}^d} \left(\frac{1}{2}\|\mathbf{x}_0 - \mathbf{y}\|_2^2 - \lambda \log p(\mathbf{y})\right), \tag{15}$$

where $\lambda > 0$ is a scalar that balances the fidelity to the input $\mathbf{x}_0$ against adherence to the distribution $p(\mathbf{y})$.

A.2. Direct Connection to MAP Estimation

We now demonstrate that the soft projection in Equation (15) is precisely equivalent to a Maximum A Posteriori (MAP) estimate, given a specific choice of the likelihood function.

Given a prior belief $p(\mathbf{y})$ and a likelihood function $p(\mathbf{x}_0|\mathbf{y})$, the MAP estimate is found by maximizing the posterior distribution $p(\mathbf{y}|\mathbf{x}_0)$ with respect to $\mathbf{y}$:

$$\begin{aligned}
\mathbf{y}_{\text{MAP}} &= \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}_0) \\
&= \arg\max_{\mathbf{y}} p(\mathbf{x}_0|\mathbf{y})p(\mathbf{y}) \quad \text{(Ignoring constant } p(\mathbf{x}_0)\text{)} \\
&= \arg\max_{\mathbf{y}} \log(p(\mathbf{x}_0|\mathbf{y})) + \log(p(\mathbf{y})) \\
&= \arg\min_{\mathbf{y}} \left(-\log p(\mathbf{x}_0|\mathbf{y}) - \log p(\mathbf{y})\right).
\end{aligned} \tag{16}$$

The soft projection in Eq. (15) aligns with the MAP objective from Eq. (16) by identifying the quadratic loss term ($\frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2$) as the negative log-likelihood ($-\log p(\mathbf{x}|\mathbf{y})$), and the penalized negative log-probability ($-\lambda \log p(\mathbf{y})$) as the scaled negative log-prior ($-\log p(\mathbf{y})$). This relationship is formally established by assuming a Gaussian likelihood:

$$\begin{aligned}
p(\mathbf{x}_0|\mathbf{y}) &= \mathcal{N}(\mathbf{x}_0|\mathbf{y}, \sigma^2 I) \\
&= \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x}_0 - \mathbf{y}\|_2^2\right)
\end{aligned} \tag{17}$$

The negative log-likelihood is then:

$$-\log p(\mathbf{x}_0|\mathbf{y}) = \frac{d}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}\|\mathbf{x}_0 - \mathbf{y}\|_2^2$$

Substituting this into the MAP objective (Equation (16)) and ignoring the constant term $\frac{d}{2}\log(2\pi\sigma^2)$, we get:

$$\mathbf{y}_{\text{MAP}} = \arg\min_{\mathbf{y}} \left(\frac{1}{2\sigma^2}\|\mathbf{x}_0 - \mathbf{y}\|_2^2 - \log p(\mathbf{y})\right). \tag{18}$$

To match this form exactly to the soft projection in Eq. (15), we can multiply the entire objective function by the constant σ^2. Since multiplication by a positive constant does not change the minimizer $\mathbf{y}_{\text{MAP}}$, we have:

$$\mathbf{y}_{\text{MAP}} = \arg\min_{\mathbf{y}} \left(\frac{1}{2}\|\mathbf{x}_0 - \mathbf{y}\|_2^2 - \sigma^2 \log p(\mathbf{y})\right). \tag{19}$$

By comparing this final expression for $\mathbf{y}_{\text{MAP}}$ to the soft projection $\mathbf{y}^*$ of Equation (15), we establish the direct correspondence:

$$\lambda = \sigma^2.$$

The parameter λ in the soft projection is therefore equivalent to the variance of the assumed Gaussian likelihood. The general form of our objective is an unnormalized negative log-posterior:

$$\arg\min_{\mathbf{y}} \left(\underbrace{[-\log p(\mathbf{x}_0|\mathbf{y})]}_{\text{Fidelity term}} + \lambda \underbrace{[-\log p(\mathbf{y})]}_{\text{Prior term}}\right). \tag{20}$$

A.3. Gaussian Prior: Solution via Linear Shrinkage

If we assume a zero-mean isotropic Gaussian prior with unit variance, the soft projection (MAP estimate) admits a closed-form solution.

Let the prior be $p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{0}, I)$. The negative log-prior term, ignoring constant terms, is:

$$-\log p(\mathbf{y}) \propto \frac{1}{2}\|\mathbf{y} - \mathbf{0}\|_2^2 = \frac{1}{2}\|\mathbf{y}\|_2^2$$

Substituting this into the soft projection objective (Eq. (15)):

$$\begin{aligned}\mathbf{y}^* &= \arg\min_{\mathbf{y}} \left(\frac{1}{2}\|\mathbf{x}_0 - \mathbf{y}\|_2^2 - \lambda \log p(\mathbf{y}) \right) \\ &= \arg\min_{\mathbf{y}} \left(\frac{1}{2}\|\mathbf{x}_0 - \mathbf{y}\|_2^2 + \lambda \left[\frac{1}{2}\|\mathbf{y}\|_2^2 \right] \right) \\ &= \arg\min_{\mathbf{y}} \left(\frac{1}{2}\|\mathbf{x}_0 - \mathbf{y}\|_2^2 + \frac{\lambda}{2}\|\mathbf{y}\|_2^2 \right).\end{aligned} \tag{21}$$

To find the optimum $\mathbf{y}^*$, we take the gradient with respect to $\mathbf{y}$ and set it to zero:

$$\nabla_{\mathbf{y}} \mathcal{L}(\mathbf{y}) = -(\mathbf{x}_0 - \mathbf{y}) + \lambda \mathbf{y} = \mathbf{0}$$

Solving for $\mathbf{y}$:

$$\mathbf{y} + \lambda \mathbf{y} = \mathbf{x}_0 \iff \mathbf{y}(1 + \lambda) = \mathbf{x}_0$$

This yields the optimal solution from **??**:

$$\mathbf{y}^* = \frac{1}{1 + \lambda}\mathbf{x}_0.$$

Recalling that $\lambda = \sigma^2$ (the likelihood variance), we substitute this back:

$$\mathbf{y}^* = \frac{1}{1 + \sigma^2}\mathbf{x}_0. \tag{22}$$

This solution demonstrates that the soft projection (or MAP estimate) is equivalent to a *linear interpolation* (or shrinkage) of the initial sample $\mathbf{x}_0$ towards the prior mean $\mathbf{0}$. The shrinkage factor, $\frac{1}{1+\sigma^2}$, depends purely on the likelihood variance (σ^2), as the prior variance is fixed at 1. A higher likelihood variance σ^2 (meaning less certainty in $\mathbf{x}_0$) results in more shrinkage toward $\mathbf{0}$.

B. Bethe-Kikuchi expansion

B.1. Derivation of the Bethe Entropy Approximation

We begin with the density function of our empirical distribution $P_{\hat{\mathbf{x}}}(\mathbf{x})$, modeled as a pairwise MRF over the pixel lattice $\mathcal{V}$:

$$p_{\mathbf{y}}(\mathbf{x}) = \frac{1}{Z} \prod_{i \in \mathcal{V}} \psi_i(x_i) \prod_{(i,j) \in \mathcal{E}} \psi_{ij}(x_i, x_j) \tag{23}$$

The true differential entropy of this joint distribution is the negative expected log-density:

$$\begin{aligned}H(P_{\mathbf{y}}) &= -\mathbb{E}_{P_{\mathbf{y}}}[\log p_{\mathbf{y}}(\mathbf{x})] \\ &= \log Z - \sum_{i \in \mathcal{V}} \mathbb{E}_{P_{\mathbf{y}}}[\log \psi_i(x_i)] - \sum_{(i,j) \in \mathcal{E}} \mathbb{E}_{P_{\mathbf{y}}}[\log \psi_{ij}(x_i, x_j)]\end{aligned} \tag{24}$$

Because calculating the global partition function Z involves an intractable high-dimensional integral over all possible pixel configurations, exactly computing this entropy is impossible.

To resolve this, we utilize the Bethe-Kikuchi approximation. The foundational insight of the Bethe approximation is that if the graph defining the MRF were a strict **tree** (containing no loops), the joint distribution can be rewritten exactly in terms of its true local marginals—the 1D pixel marginals $p_i(x_i)$ and the 2D pairwise marginals $p_{ij}(x_i, x_j)$—without needing potentials or Z.

For a tree-structured graph, the joint distribution factorizes as:

$$p_{\text{tree}}(\mathbf{x}) = \prod_{(i,j) \in \mathcal{E}} \frac{p_{ij}(x_i, x_j)}{p_i(x_i) p_j(x_j)} \prod_{i \in \mathcal{V}} p_i(x_i) \tag{25}$$

By grouping the denominator terms by node, we can count how many times each node's marginal appears. A node i appears in the denominator exactly d_i times, where d_i is the degree (number of neighbors) of node i. Thus, the exact tree factorization can be compactly written as:

$$p_{\text{tree}}(\mathbf{x}) = \prod_{(i,j) \in \mathcal{E}} p_{ij}(x_i, x_j) \prod_{i \in \mathcal{V}} \big(p_i(x_i)\big)^{1 - d_i} \tag{26}$$

The Bethe approximation posits that this tree-exact factorization serves as a robust approximation for loopy graphs (like our 2D image lattice). Substituting this approximation into the definition of entropy, we get:

$$\begin{aligned}H(P_{\mathbf{y}}) &\approx -\mathbb{E}_{P_{\mathbf{y}}} \left[\log \left(\prod_{(i,j) \in \mathcal{E}} p_{ij}(x_i, x_j) \prod_{i \in \mathcal{V}} \big(p_i(x_i)\big)^{1 - d_i} \right) \right] \\ &= -\sum_{(i,j) \in \mathcal{E}} \mathbb{E}_{P_{\mathbf{y}}}[\log p_{ij}(x_i, x_j)] - \sum_{i \in \mathcal{V}} (1 - d_i) \mathbb{E}_{P_{\mathbf{y}}}[\log p_i(x_i)]\end{aligned} \tag{27}$$

Because the expectation of a local function over the full joint distribution $P_{\mathbf{y}}$ is exactly equivalent to the expectation over its local marginal distribution, this simplifies beautifully to a sum of local entropies:

$$H(P_{\mathbf{y}}) \approx \sum_{(i,j) \in \mathcal{E}} H(P_{\mathbf{y},ij}) + \sum_{i \in \mathcal{V}} (1 - d_i) H(P_{\mathbf{y},i}) \tag{28}$$

where $H(P_{\mathbf{y},ij})$ is the joint entropy of a pixel pair, and $H(P_{\mathbf{y},i})$ is the marginal entropy of a single pixel. This successfully replaces the intractable global calculation with a scalable sum over local cliques, forming the basis of our tractable KL divergence objective.

B.2. Validation of the Bethe approximation

To verify that the proposed formulation is able to approximate the full-domain divergence with our Markov Random Field assumption and to demonstrate that the Bethe correction produces a more accurate approximation than simple one-dimensional histogram matching, we designed an analytic validation experiment based on multivariate Gaussian distributions. Consider two zero-mean Gaussian fields $G_1 = \mathcal{N}(0, \Sigma_{G_1})$ and $G_2 = \mathcal{N}(0, \Sigma_{G_2})$, defined over a two-dimensional grid of size $(N \times N)$. Both covariances are obtained as the inverses of structured precision matrices

$$\Lambda_{G_1} = a_{G_1}\mathbb{I} + b_{G_1}\mathbb{L}, \qquad \Lambda_{G_2} = a_{G_2}\mathbb{I} + b_{G_2}\mathbb{L}, \quad (29)$$

where $\mathbb{I}$ is the identity matrix and $\mathbb{L}$ is the 4-neighborhood discrete Laplacian. The parameter (a) controls the overall variance (isotropic precision), while (b) modulates the spatial correlation between neighboring pixels. Increasing (b) enforces smoother samples by strengthening local coupling.

For two zero-mean Gaussians, the Kullback–Leibler divergence has the analytical form

$$D(G_1||G_2) = \tfrac{1}{2}\left[\mathrm{tr}(\Lambda_{G_1}\Sigma_{G_2}) - d + \log\frac{\det\Sigma_{G_1}}{\det\Sigma_{G_2}}\right], \quad (30)$$

where $d = N^2$ is the dimensionality and $\Lambda_{G_1} = \Sigma_{G_1}^{-1}$. Eq. (30) computes the ground-truth divergence between two full-domain Gaussian distributions parametrized by Eq. (29). To test whether local subset divergences can approximate the global divergence, we set G_1 to be a standard i.i.d. Gaussian field (*i.e.* $a_{G_1} = 1$, $b_{G_1} = 0$) and vary G_2. In Table (3), we show different ways to compute the KL-divergence between G_1 and G_2: the analytical closed form solution of Equation (30); the unary 1D lower bound which is the sum of individual pixel divergences, which is guaranteed by information monotonicity to be a lower bound on the full KL; the 2D histogram matching without Bethe correction ($\gamma = 0$); and the Bethe correction implementation of Equation (5).

N^2	b_{G_2}	Analytic KL	Unary Err. (%)	Pairs Err. (%)	Bethe Err. (%)
32	0.5	10.6284	27.198	214.319	2.599
32	1.0	97.682	25.901	218.214	3.039
32	10.0	173.455	22.449	229.316	5.038

Table 3. **KL Divergence Analysis**.

The results presented in Table (3) show that as b_{G_2} increases, the analytic KL grows (from 10.63 to 173.46), reflecting the larger mismatch in spatial correlations between the two distributions. The unary estimator, which only matches per-pixel variances, consistently underestimates the true KL by about (22%-27%); its relative error decreases slightly as b_{G_2} grows, since a larger fraction of the total divergence is already explained by marginal variance differences. In contrast, the pairs-only estimator, which sums 2D neighbor divergences without correcting for overlap, severely overestimates the KL by more than a factor of three (214%–229% relative error), and this bias increases with stronger correlations due to systematic over-counting of shared pixels. Our Bethe-corrected estimator remains accurate across all settings: its relative error is below 3% for moderate correlations ($b_{G_2} = 0.5$) and stays within about 5% even for very strong coupling ($b_{G_2} = 10$). This supports the claim that (i) naïve 2D histogram matching is unreliable unless over-counting is corrected, and (ii) the Bethe formulation provides a robust and quantitatively accurate approximation to the full-domain divergence.

C. Implementation Details

Our final loss is shown in Equation (11):

$$\mathcal{L}_{\text{full}}(\mathbf{y}) = \sum_{k=0}^{L-1} \alpha_k \, \mathcal{L}(\mathbf{y}_k) \quad (31)$$

where $\mathbf{y}_k = \texttt{avg\_pool2d}(\mathbf{y}, \texttt{factor} = 2^k) \cdot 2^k$. In all of our experiments, we set $L = 3$ and $\alpha_0 = 1.0$, $\alpha_1 = 0.5$, $\alpha_2 = 0.25$, which we found to work the best.

D. Reward Alignment Tasks

In this section, we provide additional information regarding the reward alignment experiments of Section 5.

D.1. Baseline Implementation

All baselines follow the original implementation except for Hwang et al. [21], which we re-implemented based on the authors' description of their method.

Additionally, for methods involving multiple loss terms such as Pix2Pix-Zero [44], ReNoise [15] and Hwang *et al.* [21], we use the relative weights between the terms used by the authors:

Method	Loss	Weights
Pix2Pix-Zero [44]	$\mathcal{L}_{\text{pair}} + \lambda\mathcal{L}_{\text{KL}}$	$\lambda = 1$
ReNoise [15]	$\lambda_1\mathcal{L}_{\text{pair}} + \lambda_2\mathcal{L}_{\text{patch-KL}}$	$\lambda_1 = 10.0,\ \lambda_2 = 0.05$
Hwang *et al.* [21]	$\mathcal{L}_1 + \mathcal{L}_2 + \lambda\mathcal{L}_{\text{power}}$	$\lambda = 25.0$

Table 4. **Hyperparameters used for the baselines**.

D.2. Gradient Normalization

To ensure fairer comparisons between baselines with vastly different gradient magnitudes, we rescale them relative to each other in all our experiments. The scales are estimated by running gradient descent with each baseline on a set of 140 images from the PIE-bench dataset. Specifically, we compute the average gradient norm over these 140 images and the first 100 steps of optimization, using a learning rate of 10^{-3}. The scales are then computed such that the average estimated gradient norm remains constant across all baselines. Table 5 shows the estimated scales used in our experiment.

Method	Relative weight ω
KL [27]	4700.0
ReNO [13]	1.0
Pix2Pix-Zero [44]	3.0
ReNoise [15]	4.0
Hwang *et al*. [21]	125.0
Ours	460.0

Table 5. **Relative weights from gradient normalization**.

D.3. Aesthetic Image Generation

We evaluated aesthetic image generation by optimizing the LAION-Aesthetics Predictor V2 reward function. The results were subsequently assessed using a set of held-out metrics designed to evaluate both image quality and prompt adherence: CLIP, HPSv2, Image Reward, and PickScore.

For evaluation, we generated four images for each of the 45 animal prompts taken from the DDPO dataset [6]. The optimization process consisted of 200 steps using the SGD optimizer with a learning rate of 5.0. We applied a gradient clipping of 0.5, and both the aesthetic loss and regularization terms were weighted equally at 0.1.

We performed experiments using both the SD-Turbo and SDXL-Turbo (one-step) models, testing two distinct prompt variants:

1. "A/An [name of the animal]"
2. "A photo of a/an [name of the animal]"

The quantitative results are presented in Table 7, and additional qualitative comparisons for both SD-Turbo and SDXL-Turbo can be found in Figures 15 and 16. Our proposed regularization method significantly outperforms prior work, demonstrating its effectiveness by more successfully preserving the latent white noise structure during optimization. Crucially, it achieves on-par performance with the recently published concurrent approach by Hwang *et al*. [21].

D.4. Brightness Minimization Reward

For the attribute control experiment, the reward function was defined as the average pixel value of the generated image; minimizing this value effectively drives image brightness down. As in the aesthetic generation experiment, we evaluate the results using a set of held-out metrics.

To test the model's ability to handle conflicting objectives, we generated four images for each of 12 animal prompts using the format *"A photo of a white [animal]"*. This setup is specifically designed to assess how regularization methods enable the model to maintain the original data distribution when the reward function (brightness minimization) is in clear contradiction with the prompt (white animal). Extra quantitative results for SD-Turbo and SDXL-Turbo are shown in Table 8, and additional qualitative results are available in Figure 17.

D.5. Model-Free Image Variant Generation

For the model-free image variant generation task, we leverage the known strong spatial correlation between diffusion-generated images and their corresponding noise latents. This allows us to find noise that can create variants of a target image while preserving its underlying structure. To achieve this, we use the Pearson correlation between the target image latent and the noise as a reward:

$$R(\mathbf{y};\hat{\mathbf{x}}) = \frac{\sum_i (\mathbf{y}_i - \bar{\mathbf{y}})(\hat{\mathbf{x}}_i - \bar{\hat{\mathbf{x}}})}{\sqrt{\sum_i (\mathbf{y}_i - \bar{\mathbf{y}})^2 \sum_i (\hat{\mathbf{x}}_i - \bar{\hat{\mathbf{x}}})^2}} \tag{32}$$

where $\bar{\mathbf{y}}$ and $\bar{\hat{\mathbf{x}}}$ denote the scalar means of the noise latent $\mathbf{y}$ and the target image latent $\hat{\mathbf{x}}$, respectively.

Figure 8 shows another qualitative example of this task using 12 different starting random seeds, with the top-left image serving as our target (seed 33). As shown in the second row, without optimization, each random seed generates very different images for the same prompt (*A photo of a frog*). After optimization (200 steps, lr=0.1), the resulting noise (first row) can generate variants of the target image that preserve the subject's pose. This result generalizes to different models, as shown in the middle rows. Furthermore, the noise remains sufficiently in-distribution to generate proper images when conditioned on an entirely different prompt (*A photo of a strawberry cake*).

We qualify this as "model-free" because the optimization does not require knowing the diffusion model in advance, as the reward relies on a simple correlation metric. However, the weighting between the reward and our Gaussian regularization requires careful tuning. As shown in Figure 9, placing too much relative weight on the reward pushes the noise to match the target latent directly, while too little weight keeps the noise random, preventing it from capturing the structure of the target image. In the example shown, the op-

timal range is empirically found to be [50, 75], though this value varies depending on the target image.

E. Computational Cost and Efficiency

E.1. Scaling with Latent Dimension

Although standard KDE exhibits quadratic complexity, we achieve linear complexity without quality loss by computing pairwise distances against a fixed set of 128 bins. To assess scalability, we benchmark the optimization across resolutions ranging from 32^2 to 1024^2 under identical settings. We measure both computation time and peak GPU memory usage; Table 6 reports these metrics relative to pixel count.

Metric	32×32	64×64	128×128	256×256	512×512	1024×1024
Time (ms)	5.5326 $\pm$0.1530	5.9805 $\pm$0.0256	5.9040 $\pm$0.1243	6.9101 $\pm$0.0998	15.1327 $\pm$0.3411	79.5417 $\pm$0.7855
Mem (MB)	1.92	4.30	14.71	56.36	222.94	889.28

Table 6. **Time/step and Peak Memory** wrt. latent resolution.

E.2. Comparison with Baselines

Our approach is computationally more intensive than baselines, a trade-off for performing exact histogram matching rather than matching simple statistics. In Table 1, we compared the wall-clock times by measuring GPU time via CUDA events over 500 optimization steps (post 20-step warm-up), averaged over 5 trials excluding I/O overhead.

F. Effect of learning rate and hyperparameters

When evaluating Gaussian regularization techniques, we assess their performance across four key dimensions to ensure a mathematically rigorous and fair comparison:

- **Gaussianity:** The regularizer must project latents into the typical set of a standard normal distribution. We verify this statistically (matching marginal distributions and eliminating spatial correlations) and empirically (generating high-quality, artifact-free images via a pre-trained diffusion model).
- **Convergence and Stability:** High learning rates can inject optimizer-induced stochasticity, artificially mimicking Gaussian noise and masking a loss function's true capability. For a fair evaluation, we only compare methods within stable regimes where the loss converges and the resulting noise stabilizes, ensuring the regularizer itself drives the projection.
- **Fair Hyperparameter Tuning:** Different regularization methods operate on vastly different gradient scales. Forcing a uniform learning rate misrepresents their capabilities. Instead, we tune the learning rate (η) independently for each baseline, comparing them solely at their respective optimal values that maximize Gaussianity while maintaining convergence stability.
- **Input Robustness:** A method's effectiveness depends heavily on the initial latent state. We evaluate methods on both practical inputs (e.g., image latents with 50% noise to simulate reward-guided generation) and extreme stress tests (e.g., purely clean latents or checkerboards) to assess the global robustness of the loss.

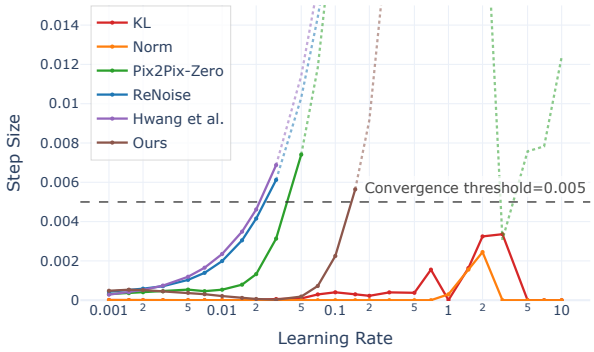

Figure 7. **Convergence Across Learning Rates.** Measured as the MSE between the last two optimized latents, step size serves as our primary indicator of optimization stability. A method's "stable regime" (solid lines) spans all learning rates until this step size strictly exceeds an acceptable threshold (dashed line). For fair evaluation, we restrict all baseline comparisons to their stable regimes, discarding diverging runs (dotted lines).

To assess the various baselines across a large range of learning rates, we created a family of datasets to cover many different inputs and noise levels. Figure 10 shows the data samples used to evaluate the methods. We quantitatively evaluate all baselines and our proposed method along the aspects mentioned above while sweeping over a large range of learning rates. The results are shown in Figures 12, 13 and 11.

G. Additional proofs

G.1. Recovering baseline proxy distributions

In this section, we formally show that minimizing the KL divergence $D_{KL}(P_{\hat{\mathbf{x}}} \,\|\, G)$ between an implicit empirical distribution $P_{\hat{\mathbf{x}}}(\mathbf{x})$ and the target standard Gaussian $G = \mathcal{N}(\mathbf{0}, \mathbf{I})$ recovers standard regularization methods when specific choices for $P_{\hat{\mathbf{x}}}(\mathbf{x})$ are made.

Let the target distribution in D dimensions be defined by the density $q(\mathbf{x})$:

$$q(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}} \exp\left(-\frac{1}{2}\|\mathbf{x}\|_2^2\right) \tag{33}$$

1. Naive (MAP Estimator)
The Naive MAP estimator implicitly assumes that the empirical distribution is a Dirac delta function centered exactly at the sample $\mathbf{y}$:

$$P_{\hat{\mathbf{x}}}(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{y}) \tag{34}$$

Plugging this into the KL divergence formula, we get:

$$\begin{aligned} D_{KL}(\delta_{\mathbf{y}} \,\|\, G) &= \int \delta(\mathbf{x}-\mathbf{y}) \log \frac{\delta(\mathbf{x}-\mathbf{y})}{q(\mathbf{x})} d\mathbf{x} \\ &= \int \delta(\mathbf{x}-\mathbf{y}) \log \delta(\mathbf{x}-\mathbf{y}) d\mathbf{x} \\ &\quad - \int \delta(\mathbf{x}-\mathbf{y}) \log q(\mathbf{x}) d\mathbf{x} \end{aligned} \tag{35}$$

The first term is the negative differential entropy of a Dirac delta. While mathematically divergent, it evaluates to an infinite constant C that is strictly independent of the optimization variable $\mathbf{y}$.

For the second term (the cross-entropy), we apply the sifting property of the Dirac delta distribution:

$$\begin{aligned} -\int \delta(\mathbf{x}-\mathbf{y}) \log q(\mathbf{x}) d\mathbf{x} &= -\log q(\mathbf{y}) \\ &= -\log\left(\frac{1}{(2\pi)^{D/2}} \exp\left(-\frac{1}{2}\|\mathbf{y}\|_2^2\right)\right) \\ &= \frac{1}{2}\|\mathbf{y}\|_2^2 + \frac{D}{2}\log(2\pi) \end{aligned} \tag{36}$$

Dropping all terms that are constant with respect to $\mathbf{y}$, the optimization objective simplifies directly to the ℓ_2 norm penalty:

$$\arg\min_{\mathbf{y}} D_{KL}(\delta_{\mathbf{y}} \,\|\, G) \equiv \arg\min_{\mathbf{y}} \frac{1}{2}\|\mathbf{y}\|_2^2 \tag{37}$$

This is the standard L_2 regularization, matching the geometric projection to the mode (the zero vector) characteristic of the MAP estimator.

2. KL Loss (1st & 2nd Moments)
Many standard approaches define the empirical distribution as a Gaussian parameterized by the sample's scalar mean $\boldsymbol{\mu}_{\mathbf{y}}$ and scalar variance $\sigma_{\mathbf{y}}^2$:

$$P_{\hat{\mathbf{x}}}(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{\mathbf{y}}, \sigma_{\mathbf{y}}^2 \mathbf{I}) \tag{38}$$

We are computing the KL divergence between two multivariate Gaussians, $P_{\hat{\mathbf{x}}} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{y}}, \sigma_{\mathbf{y}}^2\mathbf{I})$ and $G \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The analytical closed-form solution for the KL divergence between two D-dimensional Gaussians $\mathcal{N}_0(\boldsymbol{\mu}_0, \Sigma_0)$ and $\mathcal{N}_1(\boldsymbol{\mu}_1, \Sigma_1)$ is:

$$\begin{aligned} &D_{KL}(\mathcal{N}_0 \,\|\, \mathcal{N}_1) \\ &= \frac{1}{2}\left[\mathrm{tr}(\Sigma_1^{-1}\Sigma_0) + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \Sigma_1^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) - D + \log\frac{|\Sigma_1|}{|\Sigma_0|}\right] \end{aligned} \tag{39}$$

Substituting $\boldsymbol{\mu}_0 = \boldsymbol{\mu}_{\mathbf{y}}$, $\Sigma_0 = \sigma_{\mathbf{y}}^2\mathbf{I}$, $\boldsymbol{\mu}_1 = \mathbf{0}$, and $\Sigma_1 = \mathbf{I}$, we obtain:

$$D_{KL}(P_{\hat{\mathbf{x}}} \,\|\, G) = \frac{1}{2}\left[\mathrm{tr}(\sigma_{\mathbf{y}}^2\mathbf{I}) + \boldsymbol{\mu}_{\mathbf{y}}^T\mathbf{I}\boldsymbol{\mu}_{\mathbf{y}} - D + \log\frac{|\mathbf{I}|}{|\sigma_{\mathbf{y}}^2\mathbf{I}|}\right] \tag{40}$$

Using the matrix properties $\mathrm{tr}(\sigma_{\mathbf{y}}^2\mathbf{I}) = D\sigma_{\mathbf{y}}^2$ and $|\sigma_{\mathbf{y}}^2\mathbf{I}| = (\sigma_{\mathbf{y}}^2)^D$, this evaluates to:

$$\begin{aligned} D_{KL}(P_{\hat{\mathbf{x}}} \,\|\, G) &= \frac{1}{2}\left[D\sigma_{\mathbf{y}}^2 + \|\boldsymbol{\mu}_{\mathbf{y}}\|_2^2 - D - D\log(\sigma_{\mathbf{y}}^2)\right] \\ &= \frac{D}{2}\left(\sigma_{\mathbf{y}}^2 - \log(\sigma_{\mathbf{y}}^2) - 1\right) + \frac{1}{2}\|\boldsymbol{\mu}_{\mathbf{y}}\|_2^2 \end{aligned} \tag{41}$$

This perfectly matches the canonical VAE-style KL divergence loss used in prior works to align a latent's first- and second-order statistics to a standard normal prior, ignoring any spatial structures or higher-order correlations.

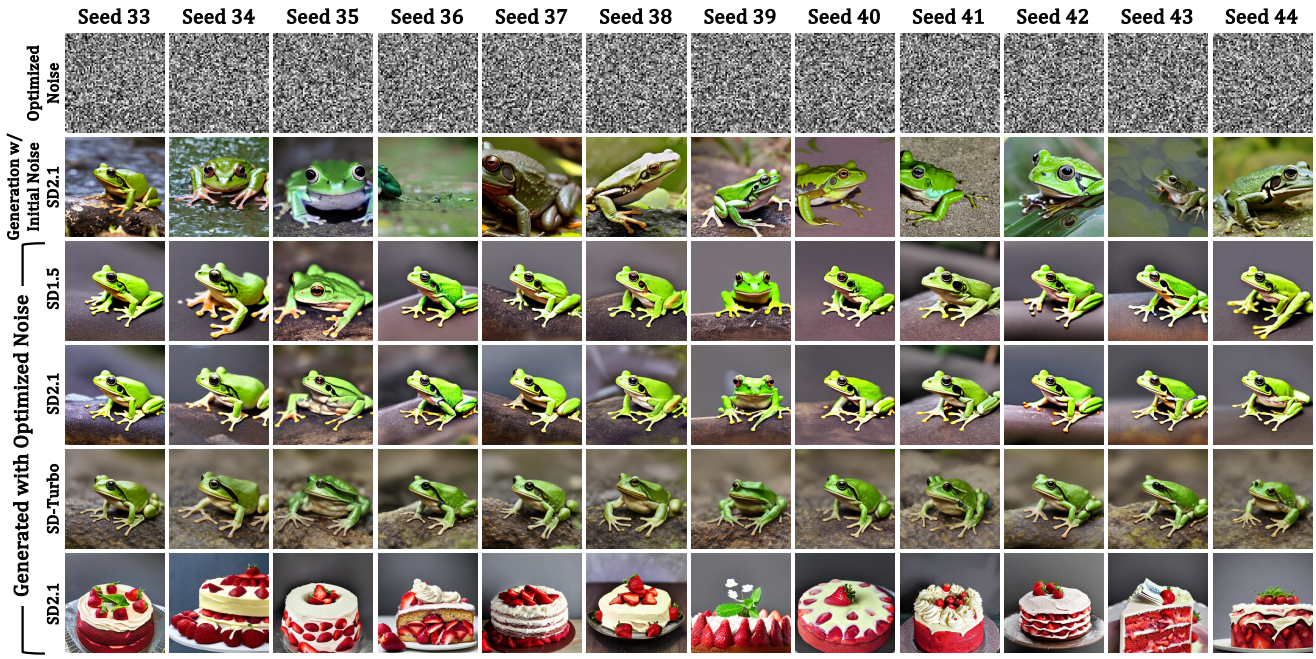

Figure 8. **More qualitative results from model-free image variant generation.**

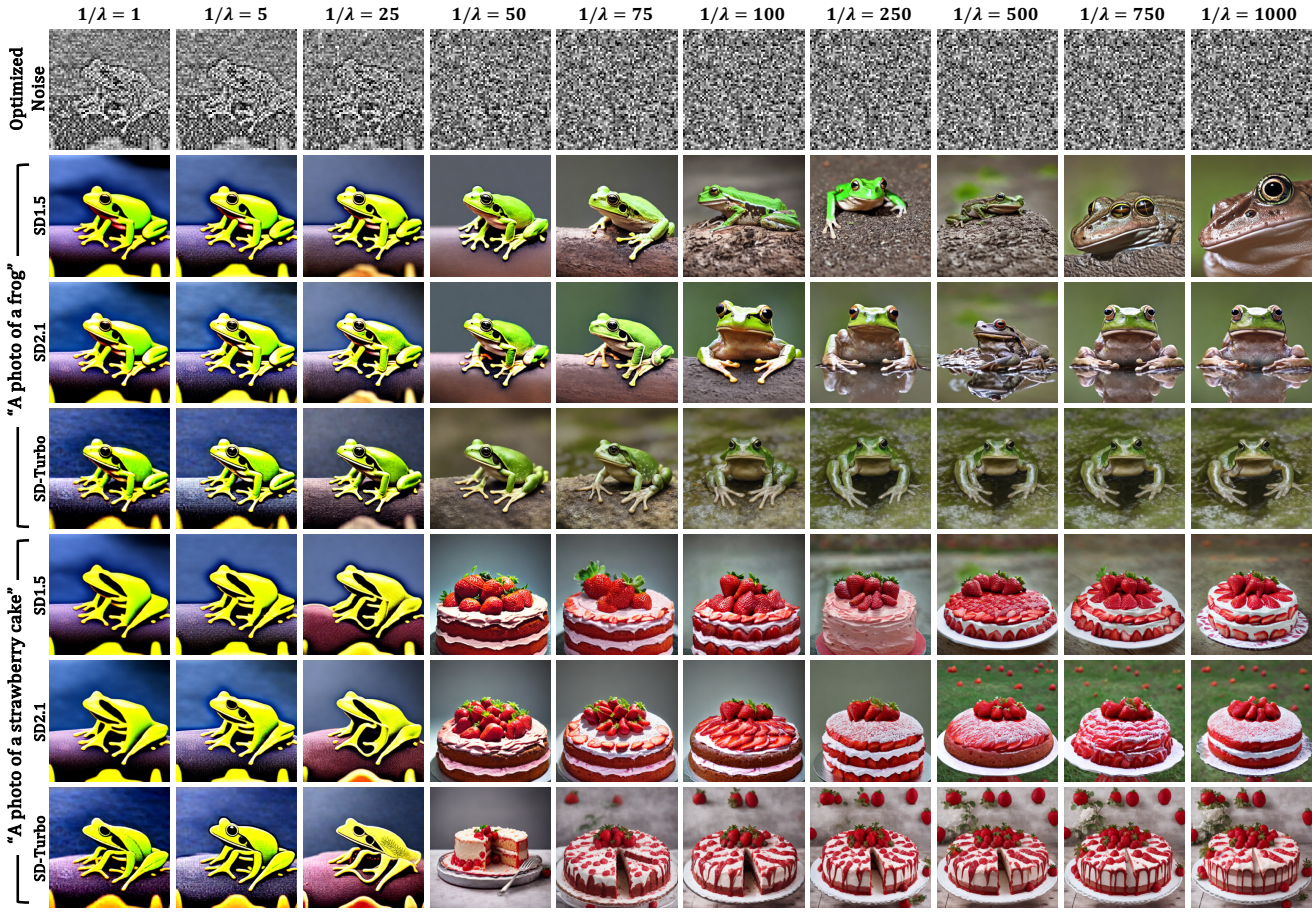

Figure 9. **Effect of relative weight between regularization and Pearson correlation reward in model-free image variant generation.**

	Prompt: "A photo of a/an [animal]"					Prompt: "A/an [animal]"				
Method	Aesth. ↑	CLIP ↑	HPSv2 ↑	ImgRwd ↑	PickSc. ↑	Aesth. ↑	CLIP ↑	HPSv2 ↑	ImgRwd ↑	PickSc. ↑
Initial	5.698	24.901	0.285	0.665	22.227	6.115	22.821	0.270	0.351	21.546
No Reg.	**8.152**	18.572	0.209	-0.831	18.770	**7.776**	17.527	0.208	-1.432	19.249
KL [27]	7.377	21.222	0.224	-0.211	19.400	7.073	19.078	0.218	-1.001	19.494
ReNO [13]	7.365	21.373	0.221	-0.326	19.389	7.076	19.008	0.218	-1.012	19.495
Pix2Pix-Zero [44]	7.395	20.601	0.231	-0.270	19.466	7.186	18.489	0.223	-0.993	19.467
ReNoise [15]	6.205	20.652	0.207	-0.607	19.039	5.858	18.730	0.203	-1.350	19.163
Hwang et al. [21]	6.241	**24.607**	0.251	0.381	20.873	5.867	**22.700**	**0.242**	**-0.220**	**20.425**
Ours	6.643	24.131	**0.268**	**0.420**	**20.945**	6.200	20.706	**0.242**	-0.330	20.137
Initial	5.440	24.997	0.292	0.822	22.543	5.986	23.826	0.290	0.876	21.960
No Reg.	**7.975**	19.682	0.209	-0.950	18.595	**7.831**	19.774	0.229	-0.767	19.605
KL [27]	6.594	22.235	0.255	0.272	19.938	6.289	21.929	0.248	-0.112	20.014
ReNO [13]	6.626	21.796	0.250	0.188	19.806	6.329	22.093	0.251	-0.053	20.075
Pix2Pix-Zero [44]	7.031	20.967	0.244	0.071	19.454	6.700	21.035	0.255	0.015	20.065
ReNoise [15]	5.806	23.286	0.257	0.248	20.385	5.694	22.381	0.245	-0.176	20.049
Hwang et al. [21]	5.863	**25.019**	0.277	0.729	21.381	5.735	**24.424**	0.274	0.706	**20.984**
Ours	6.478	24.574	**0.288**	**0.826**	**21.625**	6.198	23.499	**0.278**	**0.743**	20.920

Table 7. **Quantitative evaluation on aesthetic image generation with SD-Turbo (top block) and SDXL-Turbo (bottom block).** We use the set of animal prompts from DDPO with two variants of prompts "A photo of a/an [animal]" and "A/an [animal]".

	SD-Turbo						SDXL-Turbo					
Method	Aesth. ↑	Bright ↓	CLIP ↑	HPSv2 ↑	ImgRwd ↑	PickSc. ↑	Aesth. ↑	Bright ↓	CLIP ↑	HPSv2 ↑	ImgRwd ↑	PickSc. ↑
Initial	5.573	0.499	27.536	0.300	1.080	23.003	5.418	0.504	28.039	0.306	1.179	23.197
No Reg.	4.166	**0.155**	22.950	0.159	-1.519	18.698	3.662	0.021	18.606	0.110	-2.275	17.569
KL [27]	5.108	0.192	26.317	0.208	-0.444	20.301	4.064	**0.005**	19.360	0.101	-2.232	18.096
ReNO [13]	5.030	0.186	25.283	0.212	-0.377	20.427	4.077	0.008	20.370	0.110	-2.119	18.198
Pix2Pix-Zero [44]	5.091	0.371	25.261	0.226	-0.168	20.615	4.717	0.070	23.041	0.178	-1.579	19.020
ReNoise [15]	5.370	0.401	27.530	0.255	0.803	21.747	5.339	0.108	27.080	0.240	-0.103	20.788
Hwang et al. [21]	5.568	0.444	**27.599**	0.290	0.966	22.682	5.655	0.228	**28.169**	0.287	0.925	22.467
Ours	**5.621**	0.481	27.502	**0.297**	**1.131**	**22.896**	**5.768**	0.270	27.914	**0.298**	**1.038**	**22.823**

Table 8. **Quantitative evaluation on brightness minimization with SD-Turbo (left) and SDXL-Turbo (right).** We use a set of prompts "A photo of a white [animal]" with lr=1.0.

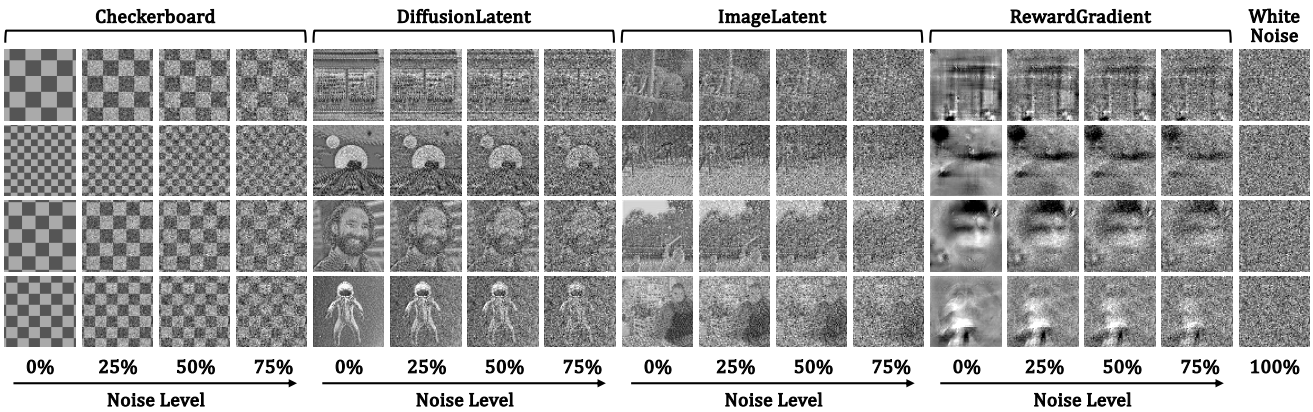

Figure 10. **Datasets for baseline evaluation.** Each column displays the samples from a distinct dataset used in our study. By varying the starting latents (patterns, natural images, diffusion outputs, and reward gradients) and noise levels, we ensure a comprehensive assessment of the methods across diverse conditions.

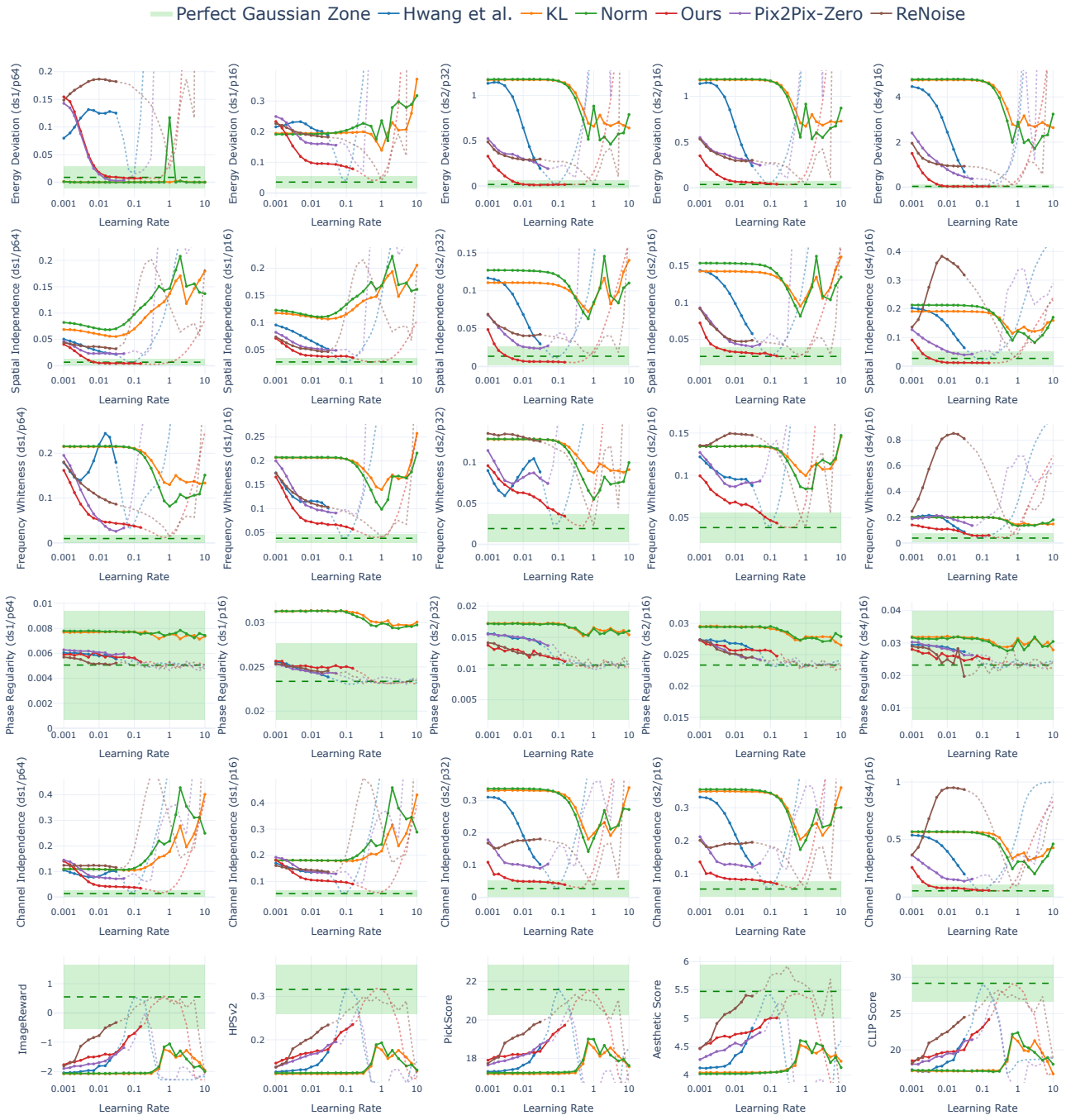

Figure 11. **Learning rate sweep behavior across baseline losses and ours, averaged over all datasets.** The top five rows track multi-scale statistical noise metrics across our spatial-scale pyramid (columns represent different downsampling and patch scales). The bottom row reports downstream image generation quality (ImageReward, HPSv2, PickScore, Aesthetic Score, and CLIP Score). The horizontal green band indicates the Perfect Gaussian Zone (baseline mean $\pm 3\sigma$). For each loss, the stable regime where the noise successfully converges is shown with a solid line, while unstable or divergent regimes are shown as dotted continuations. Crucially, while Hwang et al. (blue) peaks at $\eta = 0.1$, it is highly sensitive to learning rate changes and its peak lies within an unstable regime, implying reliance on optimizer stochasticity rather than true convergence. In contrast, our method demonstrates superior robustness across a broader range of learning rates. Even strictly within our stable convergence regime (up to $\eta = 0.15$), our approach better matches true Gaussian statistics than baselines and achieves good downstream generation quality.

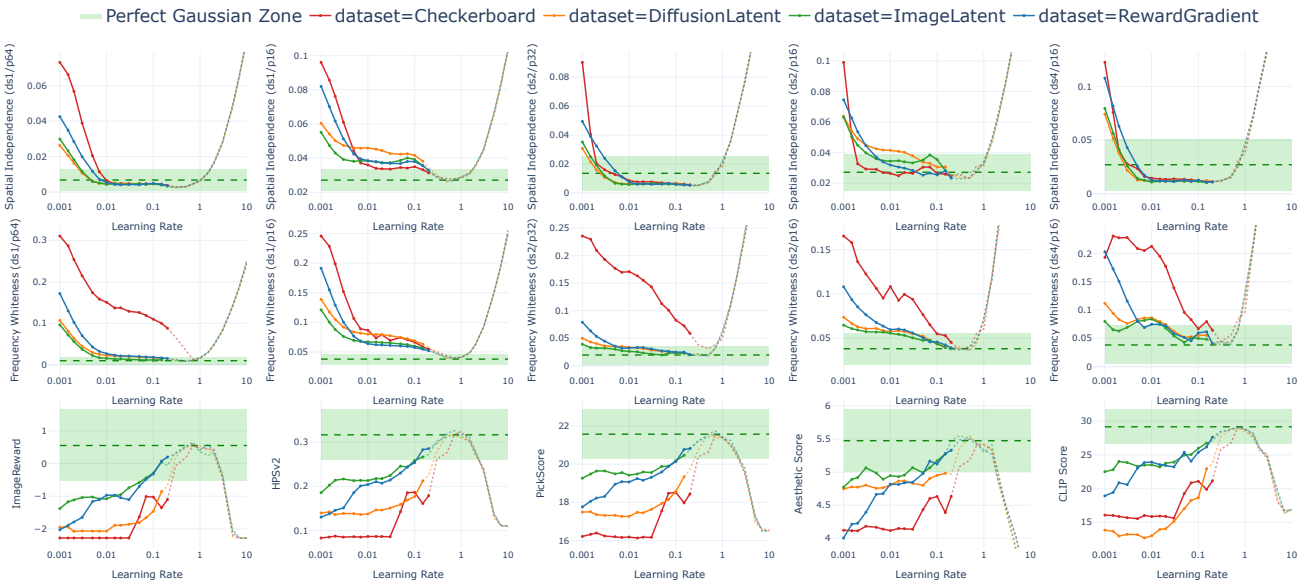

Figure 12. **Learning rate sweep behavior across different dataset types using our proposed loss, averaged over noise levels.** The top two rows evaluate multi-scale statistical noise metrics (Spatial Independence and Frequency Whiteness), while the bottom row reports downstream image generation quality. Overall, our method successfully shifts varied input distributions toward ground-truth Gaussian statistics, which facilitates improved downstream generation. However, the degree of convergence depends noticeably on the input structure. While the regularizer proves highly effective on natural image latents and reward gradients—the primary use cases for our applications—it yields more moderate improvements when applied to rigidly structured, artificial patterns such as checkerboards.

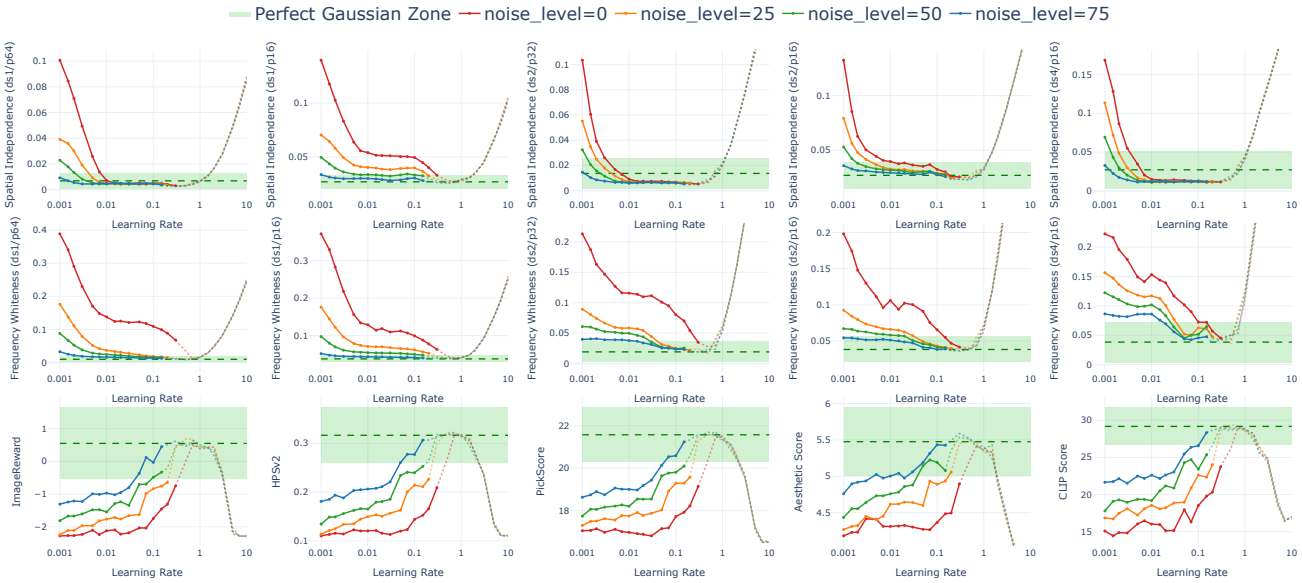

Figure 13. **Learning rate sweep behavior across different noise levels using our proposed loss, averaged over all datasets.** While inputs with higher initial noise naturally exhibit better baseline statistics, our regularization loss consistently drives the latents toward the perfect Gaussian zone across the board. Notably, this convergence is generally effective and stable as long as the input contains at least 25% initial noise.

Figure 14. **Qualitative learning rate sweep comparison.** We visualize the optimized noise latents and resulting generated images across baselines for a sample from the ImageLatent dataset (25% noise). Green frames denote outputs that fall within each method's stable convergence regime. While several baselines can produce high-quality images at specific learning rates, our method yields the most favorable visual results when strictly adhering to the required stable regime.

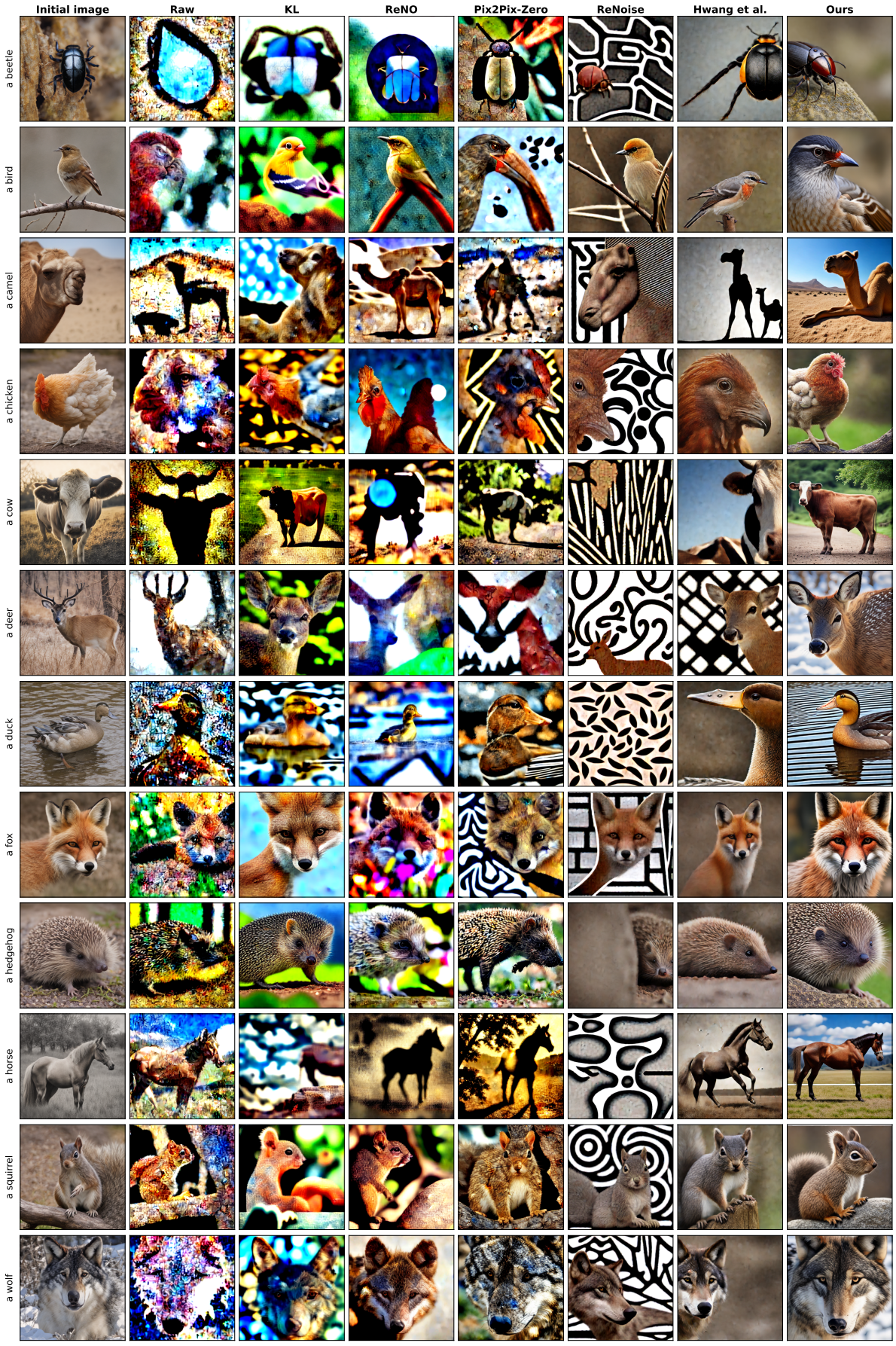

Figure 15. **More Qualitative Results from Aesthetic Image Generation with SD-Turbo**

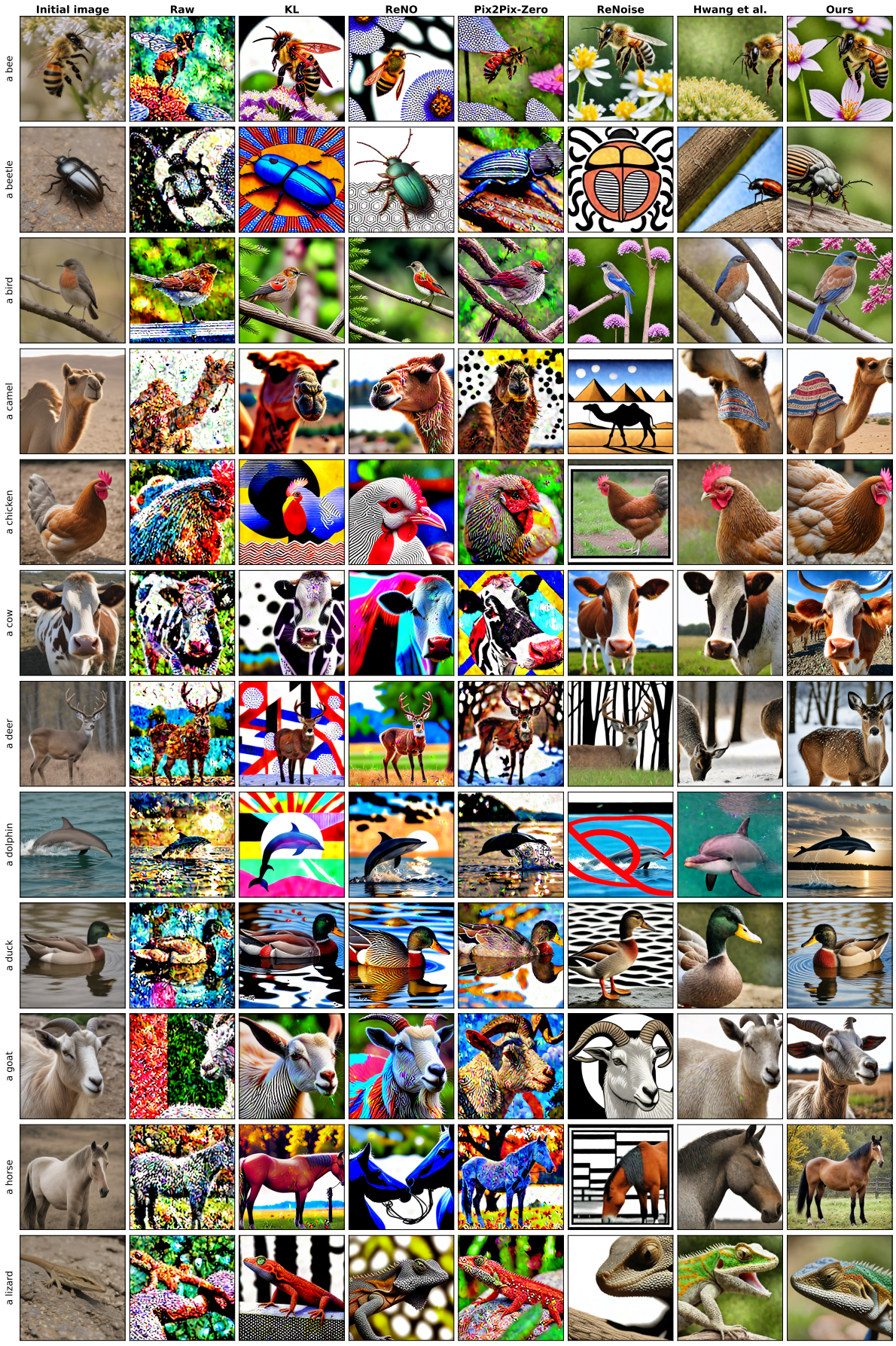

Figure 16. **More Qualitative Results from Aesthetic Image Generation with SDXL-Turbo**

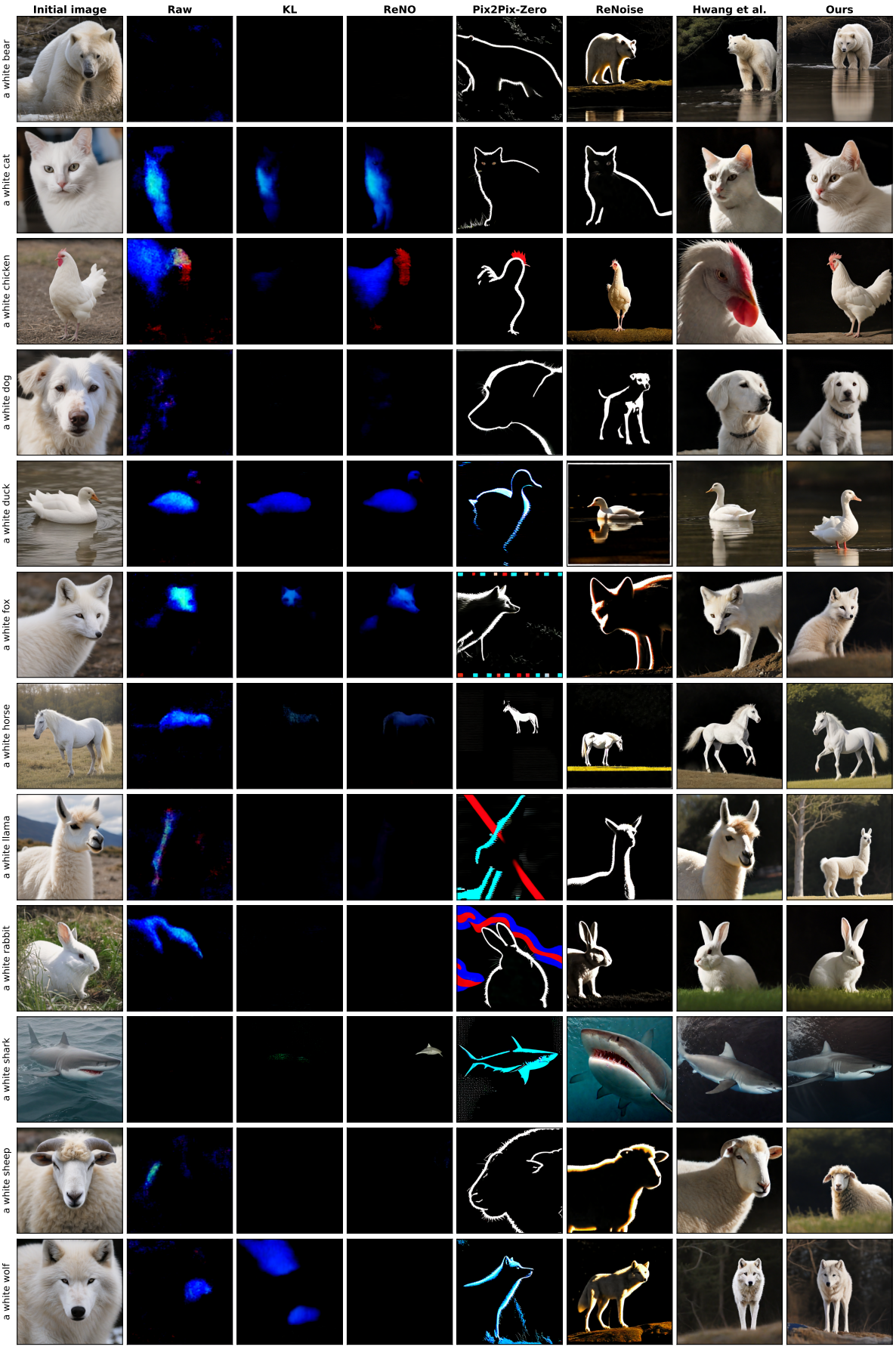

Figure 17. **More Qualitative Results from Brightness Minimization Reward with SDXL-Turbo**