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 the Maximum A Posteriori (MAP) estimate and the pointwise score $\mathcal{D}_G(\hat{\mathbf{x}}) = -\log G(\hat{\mathbf{x}})$ combined with an ℓ_2 fidelity loss $\mathcal{D}_S$, as mentioned in Section 3.1. We start by relating this to standard geometric projection.

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 $p(\hat{\mathbf{x}})$. 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:

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

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

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

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

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

where $\lambda > 0$ is a scalar that balances the fidelity to the input $\mathbf{x}_0$ against adherence to the distribution $p(\hat{\mathbf{x}})$. Notice that this directly matches the form of Equation (1) from the main text, where the fidelity term is $\mathcal{D}_S(\mathbf{x}_0, \hat{\mathbf{x}}) = \frac{1}{2}\|\mathbf{x}_0 - \hat{\mathbf{x}}\|_2^2$ and the prior compatibility term is $\mathcal{D}_G(\hat{\mathbf{x}}) = -\log p(\hat{\mathbf{x}})$, with relative weights $\lambda_S = 1$ and $\lambda_G = \lambda$.

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

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

The soft projection in Eq. (15) aligns with the MAP objective from Eq. (16) by identifying the quadratic loss term ($\frac{1}{2}\|\mathbf{x}_0 - \hat{\mathbf{x}}\|_2^2$) as the negative log-likelihood ($-\log p(\mathbf{x}_0|\hat{\mathbf{x}})$), and the penalized negative log-probability ($-\lambda \log p(\hat{\mathbf{x}})$) as the scaled negative log-prior ($-\log p(\hat{\mathbf{x}})$). This relationship is formally established by assuming a Gaussian likelihood:

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

The negative log-likelihood is then:

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

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

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

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 $\hat{\mathbf{x}}_{\text{MAP}}$, we have:

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

By comparing this final expression for $\hat{\mathbf{x}}_{\text{MAP}}$ to the soft projection $\hat{\mathbf{x}}^*$ 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:

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

Mapping this back to our notation in Equation (1), the negative log-likelihood corresponds to the fidelity term $\mathcal{D}_S(\mathbf{x}_0, \hat{\mathbf{x}})$, and the negative log-prior corresponds to the Gaussianity measure $\mathcal{D}_G(\hat{\mathbf{x}})$.

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

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

Substituting this into Equation (15):

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

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

$$\begin{aligned} \nabla_{\hat{\mathbf{x}}} \mathcal{L}(\hat{\mathbf{x}}) = -(\mathbf{x}_0 - \hat{\mathbf{x}}) + \lambda \hat{\mathbf{x}} &= \mathbf{0} \\ \implies \hat{\mathbf{x}}(1 + \lambda) &= \mathbf{x}_0 \\ \implies \hat{\mathbf{x}}^* &= \frac{1}{1+\lambda}\mathbf{x}_0. \end{aligned} \tag{22}$$

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

$$\hat{\mathbf{x}}^* = \frac{1}{1+\sigma^2}\mathbf{x}_0. \tag{23}$$

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 $\mathbf{0}$. In the context of our generic objective (Equation (1)), dividing by λ_S reveals that this effective variance corresponds precisely to the relative weighting ratio, $\sigma^2 = \lambda_G/\lambda_S$. Thus, placing a higher relative weight on the Gaussianity penalty naturally governs the magnitude of this linear shrinkage.

B. Bethe-Kikuchi Expansion

In this section, we provide the theoretical foundation for our objective function. First, we derive the tractable MRF-based entropy approximation used to regularize individual samples (Section B.1). Then, we empirically validate the accuracy of this approximation against an exact, closed-form solution on a toy Gaussian field (Section B.2).

B.1. Entropy Approximation Derivation

Pairwise MRF Factorization. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represent the pixel lattice. By the Hammersley–Clifford theorem, a strictly positive distribution satisfying the local Markov property on $\mathcal{G}$ factorizes over its cliques. For our pairwise MRF assumption, the density of the empirical distribution $P_{\hat{\mathbf{x}}}$ is defined by unary and pairwise potentials:

$$p_{\hat{\mathbf{x}}}(\mathbf{x}) = \frac{1}{Z}\prod_{i\in\mathcal{V}} \psi_i(x_i) \prod_{(i,j)\in\mathcal{E}} \psi_{ij}(x_i, x_j). \tag{24}$$

The true differential entropy is the negative expected log-density:

$$\begin{aligned} H(P_{\hat{\mathbf{x}}}) &= -\mathbb{E}_{P_{\hat{\mathbf{x}}}}[\log p_{\hat{\mathbf{x}}}(\mathbf{x})] \\ &= \log Z - \sum_{i\in\mathcal{V}} \mathbb{E}_{P_{\hat{\mathbf{x}}}}[\log \psi_i(x_i)] \\ &\quad - \sum_{(i,j)\in\mathcal{E}} \mathbb{E}_{P_{\hat{\mathbf{x}}}}[\log \psi_{ij}(x_i, x_j)]. \end{aligned} \tag{25}$$

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

Exact Tree Formulation. To resolve this intractability, we first examine the case where $\mathcal{G}$ is a strict tree (containing no loops). In this special case, the MRF potentials can be defined strictly by the true local marginals with $Z = 1$, eliminating the need for arbitrary potential functions.

Choosing an arbitrary root node, we can factorize the joint distribution using the chain rule of probability. Each node i (except the root) has exactly one parent, $\pi(i)$:

$$\begin{aligned} p_{\text{tree}}(\mathbf{x}) &= p(x_{\text{root}}) \prod_{i\neq\text{root}} p(x_i \mid x_{\pi(i)}) \\ &= p(x_{\text{root}}) \prod_{i\neq\text{root}} \frac{p_{i,\pi(i)}(x_i, x_{\pi(i)})}{p_{\pi(i)}(x_{\pi(i)})}. \end{aligned} \tag{26}$$

In this product, the numerator traverses every edge exactly once. The denominator contains every node k exactly as many times as it acts as a parent. Since a node's degree d_k in a tree equals its number of children plus one (for its own parent), node k appears in the denominator exactly $d_k - 1$

times. Rearranging terms yields the exact tree factorization:

$$p_{\text{tree}}(\mathbf{x}) = \prod_{(i,j)\in\mathcal{E}} p_{ij}(x_i, x_j) \prod_{i\in\mathcal{V}} p_i(x_i)^{1-d_i}. \tag{27}$$

The Bethe Approximation. The foundational insight of the Bethe approximation is to apply the exact tree factorization (Eq. (27)) as a robust approximation for loopy graphs, such as our 2D image lattice ($p_{\hat{\mathbf{x}}}(\mathbf{x}) \approx p_{\text{tree}}(\mathbf{x})$). Substituting this into the definition of entropy yields:

$$H(P_{\hat{\mathbf{x}}}) \approx \tag{28}$$
$$-\mathbb{E}_{P_{\hat{\mathbf{x}}}}\left[\sum_{(i,j)\in\mathcal{E}} \log p_{ij}(x_i, x_j) + \sum_{i\in\mathcal{V}} (1-d_i)\log p_i(x_i)\right].$$

By the linearity of expectation, the global expectation over $\mathbf{x}$ marginalizes directly to local expectations over the respective cliques:

$$\begin{aligned}\mathbb{E}_{P_{\hat{\mathbf{x}}}}[\log p_{ij}(x_i, x_j)] &= \mathbb{E}_{p_{ij}}[\log p_{ij}(x_i, x_j)] \\ &= -H(P_{\hat{\mathbf{x}},ij}),\end{aligned} \tag{29}$$

$$\begin{aligned}\mathbb{E}_{P_{\hat{\mathbf{x}}}}[\log p_i(x_i)] &= \mathbb{E}_{p_i}[\log p_i(x_i)] \\ &= -H(P_{\hat{\mathbf{x}},i}).\end{aligned} \tag{30}$$

Substituting these local entropies back into the summation successfully replaces the intractable global calculation with a scalable sum over local cliques:

$$H(P_{\hat{\mathbf{x}}}) \approx \sum_{(i,j)\in\mathcal{E}} H(P_{\hat{\mathbf{x}},ij}) + \sum_{i\in\mathcal{V}} (1-d_i) H(P_{\hat{\mathbf{x}},i}). \tag{31}$$

Ergodicity Assumption & Final Simplification. To materialize these distributions from a single sample, we assume the optimized sample $\hat{\mathbf{x}}$ is an ergodic spatial process. Consequently, the local marginal statistics are invariant to spatial translation across the lattice. The entropy of any individual node equals the entropy of the global 1D empirical distribution $S^{(1)}$, and the entropy of any adjacent pair equals the global 2D empirical distribution $S^{(2)}$:

$$\forall i \in \mathcal{V}, \quad H(P_{\hat{\mathbf{x}},i}) = H(P_{\hat{\mathbf{x}},S^{(1)}}), \tag{32}$$
$$\forall (i,j) \in \mathcal{E}, \quad H(P_{\hat{\mathbf{x}},ij}) = H(P_{\hat{\mathbf{x}},S^{(2)}}). \tag{33}$$

Substituting these empirical constants into Eq. (31) allows us to factor out the summations:

$$\begin{aligned}H(P_{\hat{\mathbf{x}}}) &\approx \sum_{(i,j)\in\mathcal{E}} H(P_{\hat{\mathbf{x}},S^{(2)}}) + \sum_{i\in\mathcal{V}} (1-d_i) H(P_{\hat{\mathbf{x}},S^{(1)}}) \\ &= |\mathcal{E}| H(P_{\hat{\mathbf{x}},S^{(2)}}) + \left(\sum_{i\in\mathcal{V}} (1-d_i)\right) H(P_{\hat{\mathbf{x}},S^{(1)}}).\end{aligned} \tag{34}$$

Computing the Bethe Correction (γ). By dividing Eq. (34) by the total number of edges $|\mathcal{E}|$ to normalize the objective per pairwise clique, we arrive at the final loss formulation from the main text, where γ serves as the normalized Bethe over-counting correction:

$$\gamma = \frac{1}{|\mathcal{E}|} \sum_{i\in\mathcal{V}} (1-d_i). \tag{35}$$

For an $N \times N$ grid modeled with periodic boundary conditions (a toroidal grid), every node has exactly 4 neighbors (up, down, left, right), meaning $d_i = 4$ for all N^2 nodes. By the handshaking lemma, the total number of edges is $2N^2$. The constant γ is therefore exactly analytically derived as:

$$\begin{aligned}\gamma &= \frac{1}{2N^2} \sum_{i=1}^{N^2} (1-4) \\ &= \frac{1}{2N^2}(-3N^2) = -\frac{3}{2}.\end{aligned} \tag{36}$$

Thus, to properly correct for structural over-counting on a 2D grid, the marginal 1D value entropy is weighted by exactly $-3/2$ relative to the 2D spatial entropy.

B.2. Validation on Gaussian Fields

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 $G_1 = \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{G_1})$ and $G_2 = \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{G_2})$, defined over a two-dimensional grid of size $N \times N$. Both covariances are obtained as the inverses of structured precision matrices:

$$\begin{aligned}\mathbf{\Lambda}_{G_1} &= a_{G_1}\mathbf{I} + b_{G_1}\mathbf{L}, \\ \mathbf{\Lambda}_{G_2} &= a_{G_2}\mathbf{I} + b_{G_2}\mathbf{L},\end{aligned} \tag{37}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{L}$ 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:

$$D(G_1||G_2) = \tfrac{1}{2}\left[\text{tr}(\mathbf{\Lambda}_{G_1}\mathbf{\Sigma}_{G_2}) - D + \log\frac{|\mathbf{\Sigma}_{G_1}|}{|\mathbf{\Sigma}_{G_2}|}\right], \tag{38}$$

where $D = N^2$ is the dimensionality and $\mathbf{\Lambda}_{G_1} = \mathbf{\Sigma}_{G_1}^{-1}$. Eq. (38) computes the ground-truth divergence between two full-domain Gaussian distributions parametrized

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**.

by Eq. (37). 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 (38); 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).

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):

$$\mathcal{L}_{\text{full}}(\hat{\mathbf{x}}) = \sum_{k=0}^{L-1} \alpha_k \, \mathcal{L}(\hat{\mathbf{x}}_k)\,, \tag{39}$$

where $\hat{\mathbf{x}}_k = 2^k \cdot \texttt{avg\_pool2d}(\hat{\mathbf{x}},\ \texttt{factor} = 2^k)$. This scaling factor of 2^k is mathematically required to perfectly restore unit variance after pooling, ensuring the target distribution remains $\mathcal{N}(\mathbf{0}, \mathbf{I})$ at every scale. In all of our experiments, we set $L = 3$ and $\alpha_0 = 1.0,\ \alpha_1 = 0.5,\ \alpha_2 = 0.25$, which we have found empirically to work well.

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 in the original papers:

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-to-Noise Matching

For the model-free image-to-noise matching 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:

$$R(\hat{\mathbf{x}}; \hat{\mathbf{x}}) = \frac{\sum_i (\hat{\mathbf{x}}_i - \bar{\hat{x}})(\hat{\mathbf{x}}_i - \bar{\hat{\mathbf{x}}})}{\sqrt{\sum_i (\hat{\mathbf{x}}_i - \bar{\hat{x}})^2 \sum_i (\hat{\mathbf{x}}_i - \bar{\hat{\mathbf{x}}})^2}} \tag{40}$$

where $\bar{\hat{x}}$ and $\bar{\hat{\mathbf{x}}}$ denote the scalar means of the noise latent $\hat{\mathbf{x}}$ 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. The same noise can create results across different models that live in the same space, 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 optimal range is empirically found to be (50, 75), though this value varies depending on the target image.

E. Computational Cost and Efficiency

In this section, we analyze the computational requirements of our proposed regularization method. First, we evaluate how our formulation scales with increasing latent resolutions in terms of both execution time and memory footprint. Second, we contextualize these requirements by comparing our wall-clock performance against existing baselines.

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 ±0.1530	5.9805 ±0.0256	5.9040 ±0.1243	6.9101 ±0.0998	15.1327 ±0.3411	79.5417 ±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 prior baselines, which is an expected trade-off for performing exact histogram matching rather than matching simple summary statistics. In Table 1, we compare 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. Baseline Comparisons

Evaluating noise regularization is challenging because high learning rates can inject stochasticity that artificially mimics Gaussian noise, masking the loss function's true performance. To disentangle true geometric projection from this optimizer-induced noise, we introduce a rigorous multi-scale statistical framework to benchmark our method against prior work across diverse inputs and learning rates.

F.1. Experimental Setting

To ensure a fair and comprehensive comparison, we base our evaluation on the following elements:

Input Robustness. To assess cross-domain capabilities, we evaluate on a diverse dataset of initial latents—checkerboards, natural images, diffusion-generated images, and aesthetic score gradient maps—coupled with varying levels of initial white noise (0%, 25%, 50%, 75%). Figure 10 visualizes these samples.

Multi-Scale Statistical Metrics. A successful regularizer must project the input into the typical set of the standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. We assess this using five metrics evaluated across a spatial-scale pyramid. For a given downsampling factor ds and patch size p, we pool the latent, rescale to preserve variance, and average the following metrics over non-overlapping $p \times p$ patches:

- **Energy Deviation:** Measures the deviation of the patch variance from the expected unit variance, calculated as $|\frac{1}{k}\sum x_i^2 - 1.0|$, where k is the patch size.
- **Spatial Independence:** Evaluates whether the pixel values follow the expected 1D marginal normal distribution. We measure this using the Kolmogorov-Smirnov (KS) test statistic against $\mathcal{N}(0, 1)$.
- **Frequency Whiteness:** Evaluates the power spectrum of the 2D Fast Fourier Transform (FFT). For standard white noise, the squared magnitude of the FFT coefficients should follow an Exponential distribution. We report the KS test statistic against this distribution.
- **Phase Regularity:** Evaluates the phase angles of the 2D FFT, which should be uniformly distributed in all directions. We report the KS test statistic against a uniform distribution on $[-\pi, \pi]$.
- **Channel Independence:** Measures the squared L_2 norm of the 4-channel latent vectors at each spatial location. In a standard Gaussian latent, this sum of squares should follow a Chi-squared distribution with 4 degrees of freedom (χ^2_4). We report the KS test statistic against this distribution.

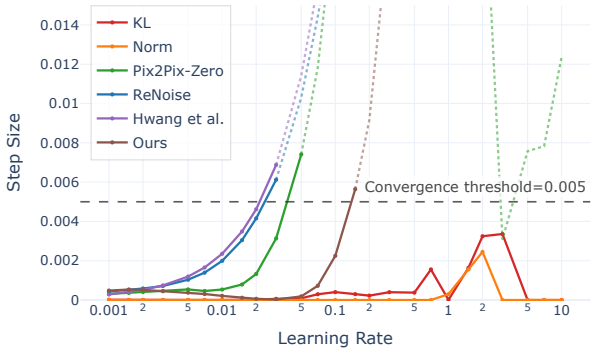

Figure 7. **Convergence Across Learning Rates.** We use the final optimization step size to identify each method's "stable regime" (solid lines), bounded by an acceptable convergence threshold (dashed line). Diverging runs (dotted lines) are discarded from our baseline comparisons.

Convergence and Stability. As previously noted, high learning rates (η) can inject stochasticity that artificially improves noise metrics. For a fair evaluation, we strictly identify the *stable regime* for each method (Figure 7). We define this by measuring the Mean Squared Error (MSE) between the latents of the final two optimization steps. A low MSE (step size) indicates that the noise has converged, proving that the regularizer itself is driving the projection. Conversely, a large step size indicates divergent or highly stochastic behavior.

Suitability for Diffusion Models. Finally, we empirically assess whether the optimized latents serve as viable inputs for image generation. We feed the optimized noises into Stable Diffusion 2.1 conditioned on the prompt *"A photo of a corgi"*. We evaluate the resulting generation quality using ImageReward, HPSv2, PickScore, and Aesthetic Score, alongside CLIP cosine similarity for prompt alignment. We compare baseline methods solely at their respective optimal learning rates that maximize Gaussianity while maintaining convergence stability.

F.2. Results & Discussion

Statistical Alignment and Robustness. Figure 11 shows the results aggregated over all datasets. Across the statistical metrics, our proposed method consistently brings the inputs closer to the standard Gaussian typical set (the target ranges indicated in green) than prior methods. Crucially, it maintains this performance over a much wider range of learning rates. While our concurrent baseline, Hwang et al. [21], performs very well at the specific learning rate used in their paper ($\eta = 0.1$), the method is highly sensitive; its performance degrades sharply as the learning rate shifts.

Optimizer Stochasticity vs. True Convergence. When examining downstream image quality, almost all methods (except KL and norm-based regularizers) can produce good images at certain learning rates. However, these peaks in visual quality often occur *outside* of the method's stable regime (indicated by dotted lines). We interpret this as strong evidence that at these high learning rates, the optimizer's stochasticity is acting as a crutch, artificially adding the noise required to yield good images. When restricted to the stable, deterministic regime, our method performs competitively against both Hwang et al. [21] and ReNoise [15].

The Low-Frequency Paradox. A notable finding in our evaluation is the behavior of ReNoise [15]. ReNoise occasionally produces very high-quality images despite exhibiting extremely poor frequency whiteness metrics (visible as the diverging blue line in the spectral plots). This discrepancy suggests that not every deviation from theoretical Gaussianity affects the diffusion denoising process equally. Specifically, it implies that modern diffusion models may be surprisingly robust to non-Gaussian statistics in the lower spatial frequencies. This empirical observation aligns with recent research on noise manipulation, which has shown that adding low frequency structure to the noise can lead to higher-quality image generation [1].

Impact of Input Pattern and Noise Level. In Figures 12 and 13, we isolate the behavior of our loss by input pattern type and initial noise level, respectively. A clear hierarchical trend emerges: highly structured, artificial patterns (like checkerboards) are significantly harder to project into the Gaussian typical set than natural image latents or reward gradients. Similarly, decreasing the initial noise level progressively pushes the input further out-of-distribution, increasing the optimization friction required to project it back. We observed identical behavioral trends across all other evaluated baselines.

Qualitative Confirmation. Finally, in Figure 14, we provide a visual progression of the optimized latents and their corresponding generated images across different baselines and learning rates. This qualitative comparison grounds our statistical findings, clearly illustrating how statistical divergence, optimizer instability, and successful geometric projection manifest in the final visual outputs.

Conclusion. Our evaluation underscores the critical need to disentangle true statistical projection from optimizer-induced artifacts. By providing competitive statistical alignment and broader optimization stability across diverse inputs, our Bethe-corrected MRF formulation proves to be a highly reliable, mathematically principled regularizer for inference-time latent optimization.

G. Recovering Baseline Proxy Distributions

In this section, we formally demonstrate that minimizing the KL divergence $D_{\mathrm{KL}}(P_{\hat{\mathbf{x}}} \,\|\, G)$ 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 proxy distributions are chosen for $P_{\hat{\mathbf{x}}}(\mathbf{x})$.

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

$$q(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}} \exp\left(-\frac{1}{2}\|\mathbf{x}\|_2^2\right). \tag{41}$$

G.1. Naive ℓ_2 Regularization (MAP Estimator)

The Naive MAP estimator implicitly assumes that the empirical distribution is a Dirac delta function centered exactly at the optimized sample $\hat{\mathbf{x}}$:

$$P_{\hat{\mathbf{x}}}(\mathbf{x}) = \delta(\mathbf{x} - \hat{\mathbf{x}}). \tag{42}$$

Plugging this into the KL divergence formula, we split the objective into two terms:

$$\begin{aligned} D_{\mathrm{KL}}(\delta_{\hat{\mathbf{x}}} \,\|\, G) &= \int \delta(\mathbf{x} - \hat{\mathbf{x}}) \log \frac{\delta(\mathbf{x} - \hat{\mathbf{x}})}{q(\mathbf{x})} d\mathbf{x} \\ &= \int \delta(\mathbf{x} - \hat{\mathbf{x}}) \log \delta(\mathbf{x} - \hat{\mathbf{x}}) d\mathbf{x} \\ &\quad - \int \delta(\mathbf{x} - \hat{\mathbf{x}}) \log q(\mathbf{x}) d\mathbf{x}. \end{aligned} \tag{43}$$

The first term represents the negative differential entropy of the Dirac delta. Formally, this is evaluated as the limit of an isotropic Gaussian as its variance approaches zero. Because this divergent term depends solely on the variance and is strictly independent of the location parameter $\hat{\mathbf{x}}$, it acts as an additive constant C with respect to our optimization variables.

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

$$\begin{aligned} -\int \delta(\mathbf{x} - \hat{\mathbf{x}}) \log q(\mathbf{x}) d\mathbf{x} &= -\log q(\hat{\mathbf{x}}) \\ &= -\log\left(\frac{\exp\left(-\frac{1}{2}\|\hat{\mathbf{x}}\|_2^2\right)}{(2\pi)^{D/2}}\right) \\ &= \frac{1}{2}\|\hat{\mathbf{x}}\|_2^2 + \frac{D}{2}\log(2\pi). \end{aligned} \tag{44}$$

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

$$\arg\min_{\hat{\mathbf{x}}} D_{\mathrm{KL}}(\delta_{\hat{\mathbf{x}}} \,\|\, G) \equiv \arg\min_{\hat{\mathbf{x}}} \frac{1}{2}\|\hat{\mathbf{x}}\|_2^2. \tag{45}$$

This perfectly recovers standard ℓ_2 regularization, matching the geometric projection to the mode (the zero vector) that is characteristic of the MAP estimator.

G.2. VAE-Style KL Loss (1st and 2nd Moments)

Many standard approaches instead define the empirical distribution as a Gaussian parameterized by the sample's scalar mean $\boldsymbol{\mu}_{\hat{\mathbf{x}}}$ and scalar variance $\sigma^2_{\hat{\mathbf{x}}}$:

$$P_{\hat{\mathbf{x}}}(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{\hat{\mathbf{x}}}, \sigma^2_{\hat{\mathbf{x}}}\mathbf{I}). \tag{46}$$

We compute the KL divergence between these two multivariate Gaussians, $P_{\hat{\mathbf{x}}} \sim \mathcal{N}(\boldsymbol{\mu}_{\hat{\mathbf{x}}}, \sigma^2_{\hat{\mathbf{x}}}\mathbf{I})$ 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:

$$\begin{aligned} D_{\mathrm{KL}}(\mathcal{N}_0 \,\|\, \mathcal{N}_1) = \frac{1}{2}\Bigg[&\mathrm{tr}(\Sigma_1^{-1}\Sigma_0) \\ &+ (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \Sigma_1^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) \\ &- D + \log\frac{|\Sigma_1|}{|\Sigma_0|}\Bigg]. \end{aligned} \tag{47}$$

Substituting our specific parameters ($\boldsymbol{\mu}_0 = \boldsymbol{\mu}_{\hat{\mathbf{x}}}$, $\Sigma_0 = \sigma^2_{\hat{\mathbf{x}}}\mathbf{I}$, $\boldsymbol{\mu}_1 = \mathbf{0}$, and $\Sigma_1 = \mathbf{I}$), we obtain:

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

Using the matrix trace property $\mathrm{tr}(\sigma^2_{\hat{\mathbf{x}}}\mathbf{I}) = D\sigma^2_{\hat{\mathbf{x}}}$ and the determinant property $|\sigma^2_{\hat{\mathbf{x}}}\mathbf{I}| = (\sigma^2_{\hat{\mathbf{x}}})^D$, this objective evaluates to:

$$\begin{aligned} D_{\mathrm{KL}}(P_{\hat{\mathbf{x}}} \,\|\, G) &= \frac{1}{2}\left[D\sigma^2_{\hat{\mathbf{x}}} + \|\boldsymbol{\mu}_{\hat{\mathbf{x}}}\|_2^2 - D - D\log(\sigma^2_{\hat{\mathbf{x}}})\right] \\ &= \frac{D}{2}\left(\sigma^2_{\hat{\mathbf{x}}} - \log(\sigma^2_{\hat{\mathbf{x}}}) - 1\right) + \frac{1}{2}\|\boldsymbol{\mu}_{\hat{\mathbf{x}}}\|_2^2. \end{aligned} \tag{48}$$

This formula exactly 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. Crucially, by strictly defining $P_{\hat{\mathbf{x}}}$ using only global moments, this formulation completely ignores any spatial structures, higher-order correlations, or local dependencies.

G.3. Norm Regularization (Hypersphere Shell)

ReNO applies regularization by minimizing the negative log-likelihood of the sample's norm. If a D-dimensional vector $\mathbf{x}$ follows a standard normal distribution $G = \mathcal{N}(\mathbf{0}, \mathbf{I})$, its L_2 norm $\|\mathbf{x}\|_2$ follows a Chi distribution with D degrees of freedom (χ_D). The density is given by:

$$p_\chi(r; D) = \frac{r^{D-1}\exp(-r^2/2)}{2^{D/2-1}\Gamma(D/2)}. \tag{49}$$

The negative log-likelihood of the norm $r = \|\hat{\mathbf{x}}\|_2$ is thus:

$$-\log p_\chi(\|\hat{\mathbf{x}}\|_2; D) = \frac{1}{2}\|\hat{\mathbf{x}}\|_2^2 - (D-1)\log\|\hat{\mathbf{x}}\|_2 + C, \tag{50}$$

where C encompasses all constant terms.

We can recover this exact objective within our framework by assuming that the empirical distribution $P_{\hat{\mathbf{x}}}$ discards all directional information from the sample, retaining only its distance from the origin. Formally, we define $P_{\hat{\mathbf{x}}}(\mathbf{x})$ as a uniform distribution over the surface of a D-dimensional hypersphere $S_{D-1}(r)$ with radius $r = \|\hat{\mathbf{x}}\|_2$.

The KL divergence is decomposed into differential entropy and cross-entropy:

$$D_{\mathrm{KL}}(P_{\hat{\mathbf{x}}} \,\|\, G) = -H(P_{\hat{\mathbf{x}}}) - \mathbb{E}_{P_{\hat{\mathbf{x}}}}[\log q(\mathbf{x})]. \tag{51}$$

First, we evaluate the differential entropy $-H(P_{\hat{\mathbf{x}}})$. For a uniform distribution defined on a domain of "volume" V (in this case, the surface area of the hypersphere), the entropy is strictly $\log V$. The surface area of a D-dimensional hypersphere of radius r is:

$$A_D(r) = \frac{2\pi^{D/2}}{\Gamma(D/2)}r^{D-1}. \tag{52}$$

Substituting $r = \|\hat{\mathbf{x}}\|_2$, the negative entropy becomes:

$$\begin{aligned} -H(P_{\hat{\mathbf{x}}}) &= -\log A_D(\|\hat{\mathbf{x}}\|_2) \\ &= -(D-1)\log\|\hat{\mathbf{x}}\|_2 - \log\left(\frac{2\pi^{D/2}}{\Gamma(D/2)}\right). \end{aligned} \tag{53}$$

Next, we evaluate the cross-entropy term. Because the target distribution G is an isotropic Gaussian, its density depends only on the norm of $\mathbf{x}$. Since our empirical distribution $P_{\hat{\mathbf{x}}}$ strictly confines $\mathbf{x}$ to the shell where $\|\mathbf{x}\|_2 = \|\hat{\mathbf{x}}\|_2$, the log-density $\log q(\mathbf{x})$ is constant everywhere on this support:

$$\begin{aligned} -\mathbb{E}_{P_{\hat{\mathbf{x}}}}[\log q(\mathbf{x})] &= -\log q(\hat{\mathbf{x}}) \\ &= \frac{1}{2}\|\hat{\mathbf{x}}\|_2^2 + \frac{D}{2}\log(2\pi). \end{aligned} \tag{54}$$

Combining the entropy (Eq. (53)) and cross-entropy (Eq. (54)) components, we obtain:

$$\begin{aligned} D_{\mathrm{KL}}(P_{\hat{\mathbf{x}}} \,\|\, G) = &\frac{1}{2}\|\hat{\mathbf{x}}\|_2^2 - (D-1)\log\|\hat{\mathbf{x}}\|_2 \\ &+ \frac{D}{2}\log(2\pi) - \log\left(\frac{2\pi^{D/2}}{\Gamma(D/2)}\right). \end{aligned} \tag{55}$$

Dropping all terms that are constant with respect to the optimization variable $\hat{\mathbf{x}}$, the objective simplifies to:

$$\begin{aligned} &\arg\min_{\hat{\mathbf{x}}} D_{\mathrm{KL}}(P_{\hat{\mathbf{x}}} \,\|\, G) \\ &\equiv \arg\min_{\hat{\mathbf{x}}}\left(\frac{1}{2}\|\hat{\mathbf{x}}\|_2^2 - (D-1)\log\|\hat{\mathbf{x}}\|_2\right). \end{aligned} \tag{56}$$

This exactly matches the norm regularization objective in Eq. (50). This proof demonstrates that constraining a latent's norm is mathematically equivalent to aligning an empirical uniform hypersphere to the Gaussian prior.

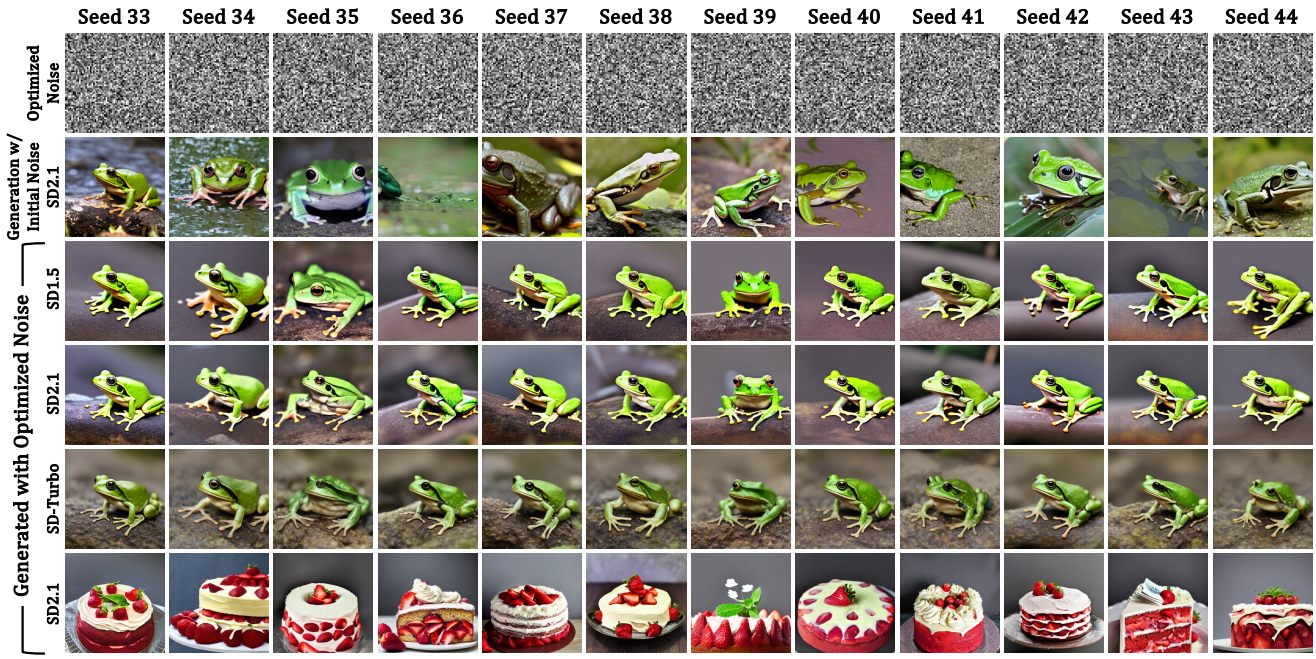

Figure 8. **More Qualitative Results from Model-Free Image-to-Noise Matching.**

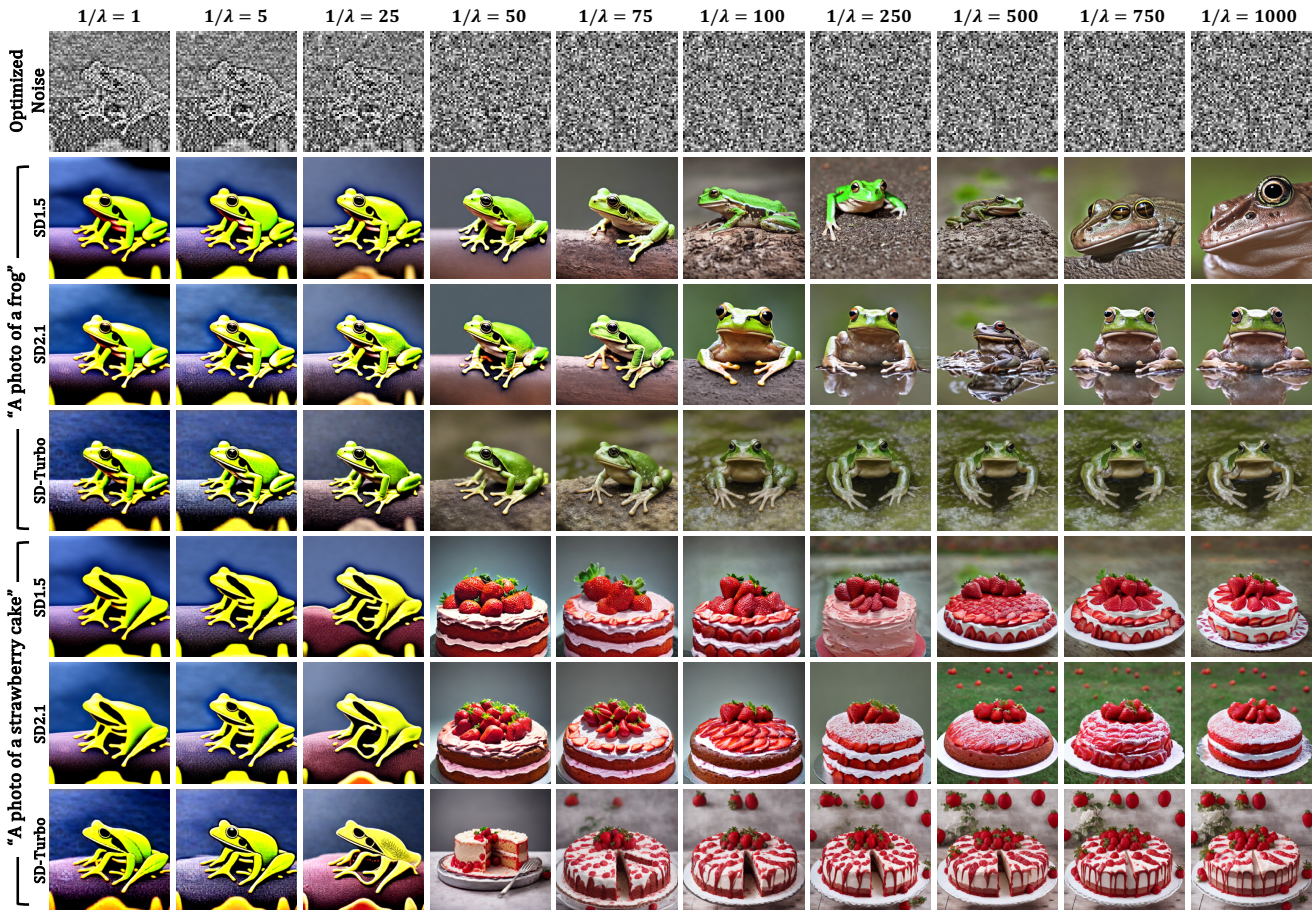

Figure 9. **Effect of Relative Weight Between Regularization and Pearson Correlation in Model-Free Image-to-Noise Matching.**

	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 learning rate $\eta = 1.0$.

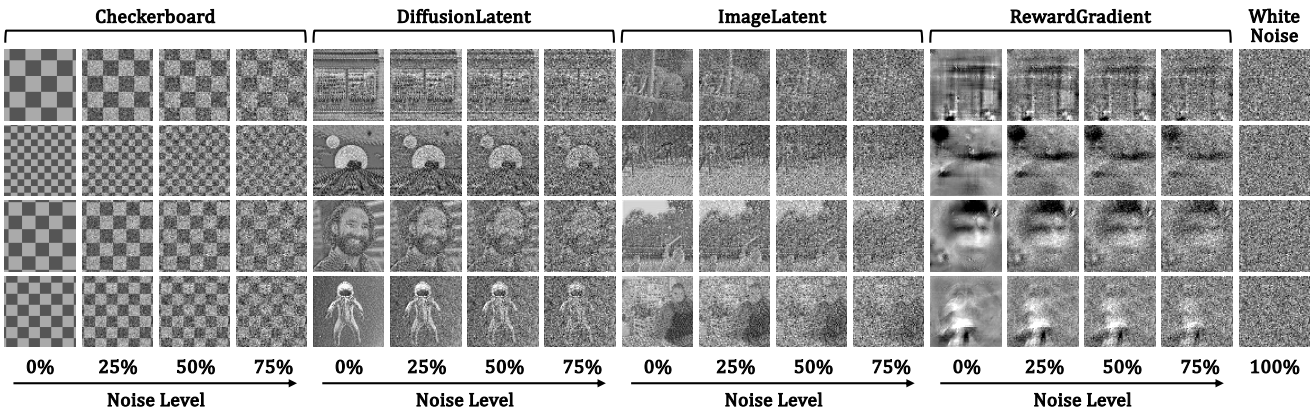

Figure 10. **Datasets for Baseline Evaluation.** Each column displays the samples from a distinct dataset used in our study. By varying the starting latents (patterns, natural images, diffusion outputs, and reward gradients) and noise levels, we ensure a comprehensive assessment of the methods across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

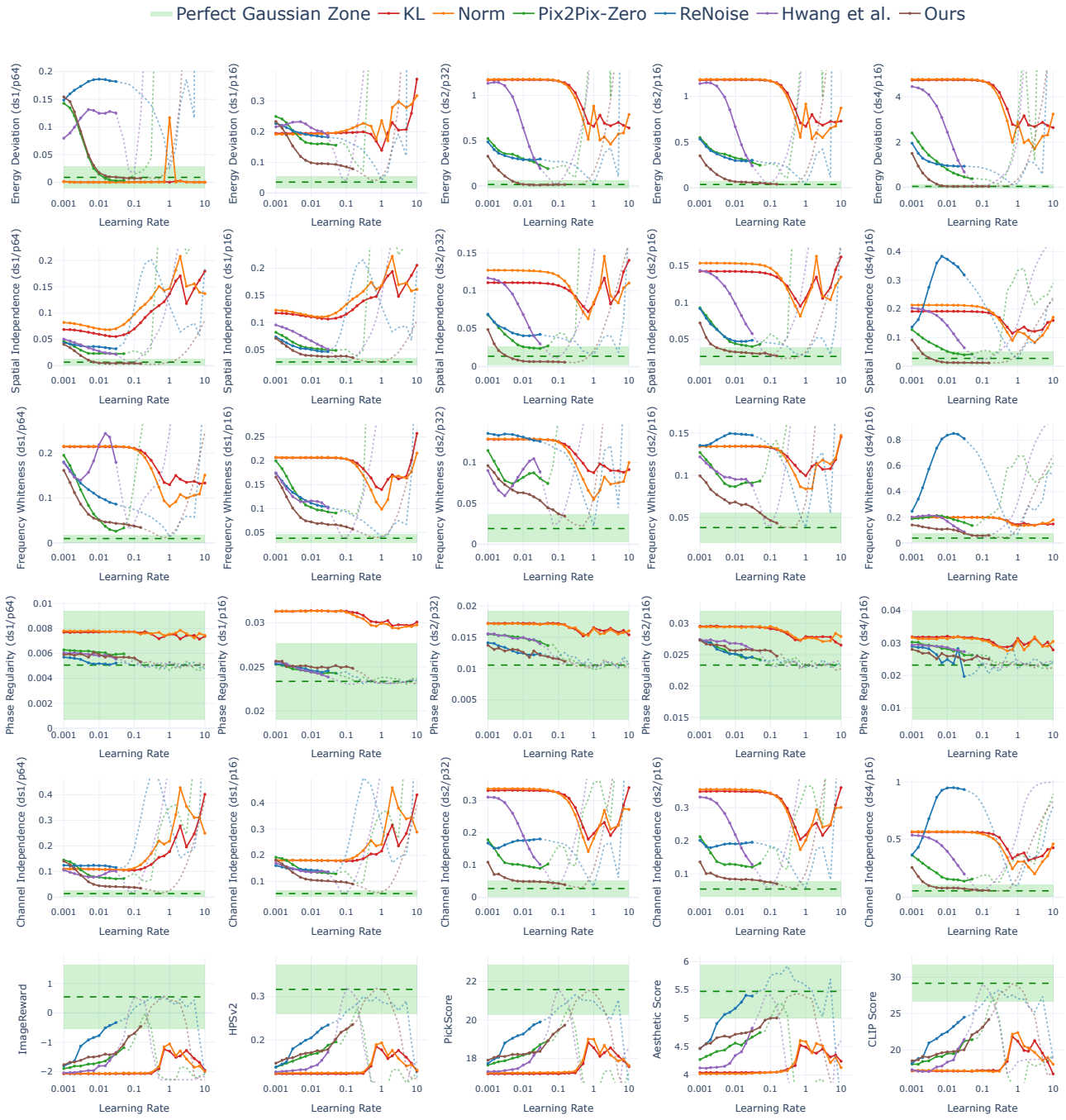

Figure 11. **Learning Rate Sweep Behavior Across Baseline Losses and Ours, Averaged Over All Datasets.** The top five rows track multi-scale statistical noise metrics across our spatial-scale pyramid (columns represent different downsampling and patch scales). The bottom row reports downstream image 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.* (purple) 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 (brown) maintains stability across a broader range of learning rates. Even strictly within our stable convergence regime (up to $\eta = 0.15$), our approach achieves competitive statistical alignment and downstream generation quality.

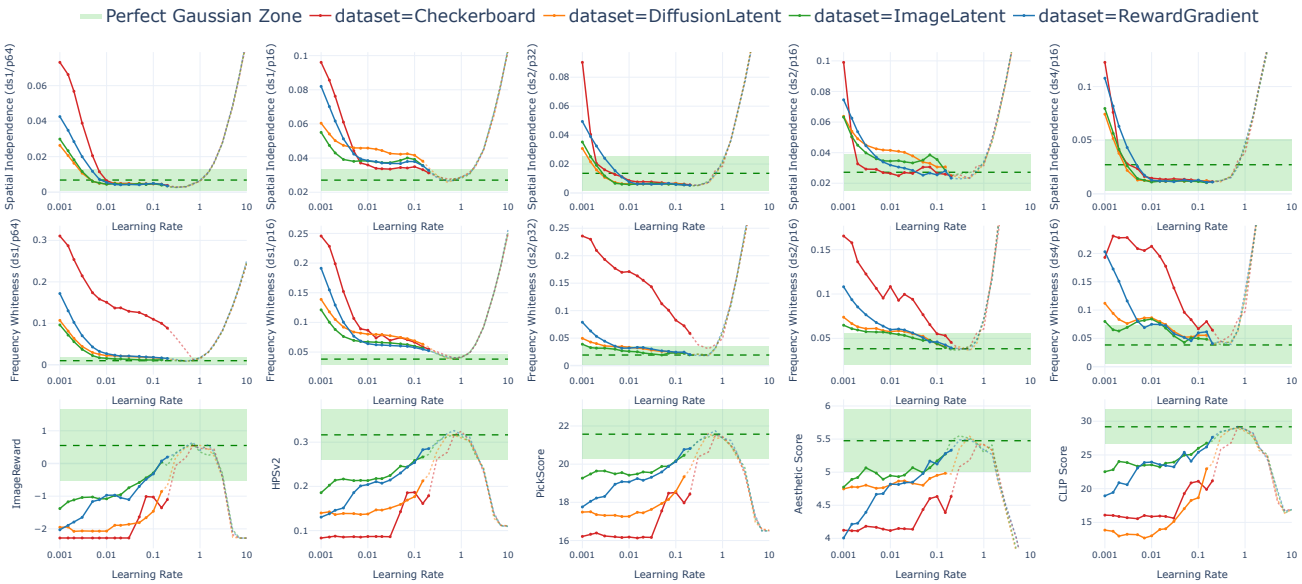

Figure 12. **Learning Rate Sweep Across Different Dataset Types Using Our Proposed Loss, Averaged Over Noise Levels.** The top two rows evaluate multi-scale statistical noise metrics (Spatial Independence and Frequency Whiteness), while the bottom row reports downstream image generation quality. Overall, 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.

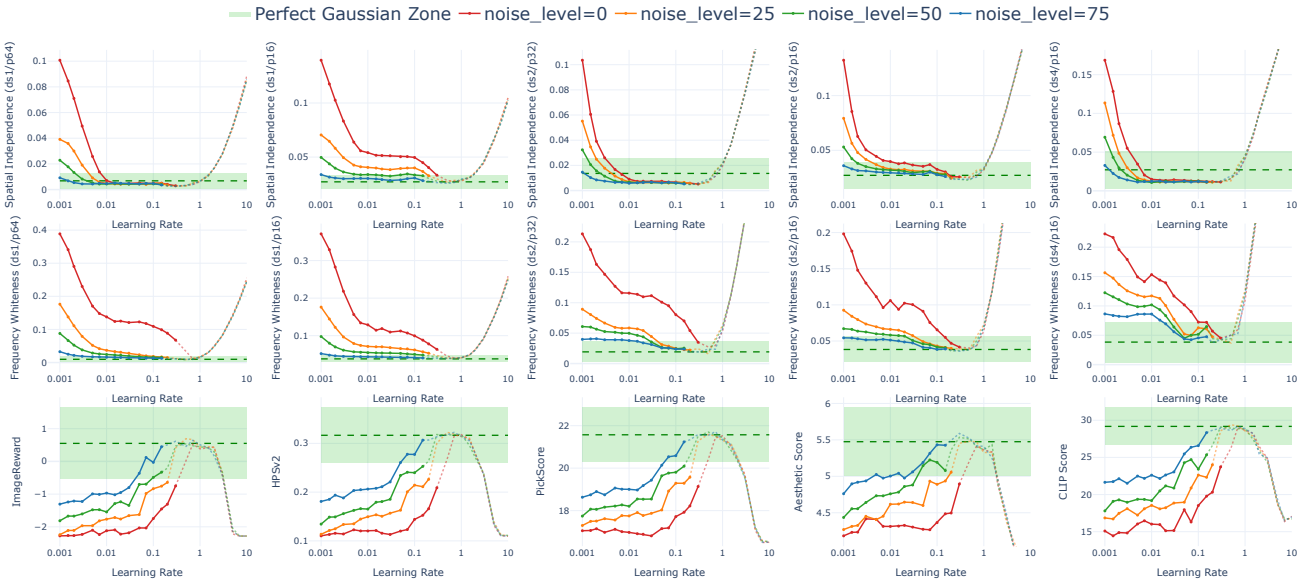

Figure 13. **Learning Rate Sweep Across Different Noise Levels Using Our Proposed Loss, Averaged Over All Datasets.** While inputs with higher initial noise naturally exhibit better baseline statistics, our regularization loss consistently drives the latents toward the Perfect Gaussian Zone across the 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 consistent good-quality visual results when strictly adhering to the required stable regime.

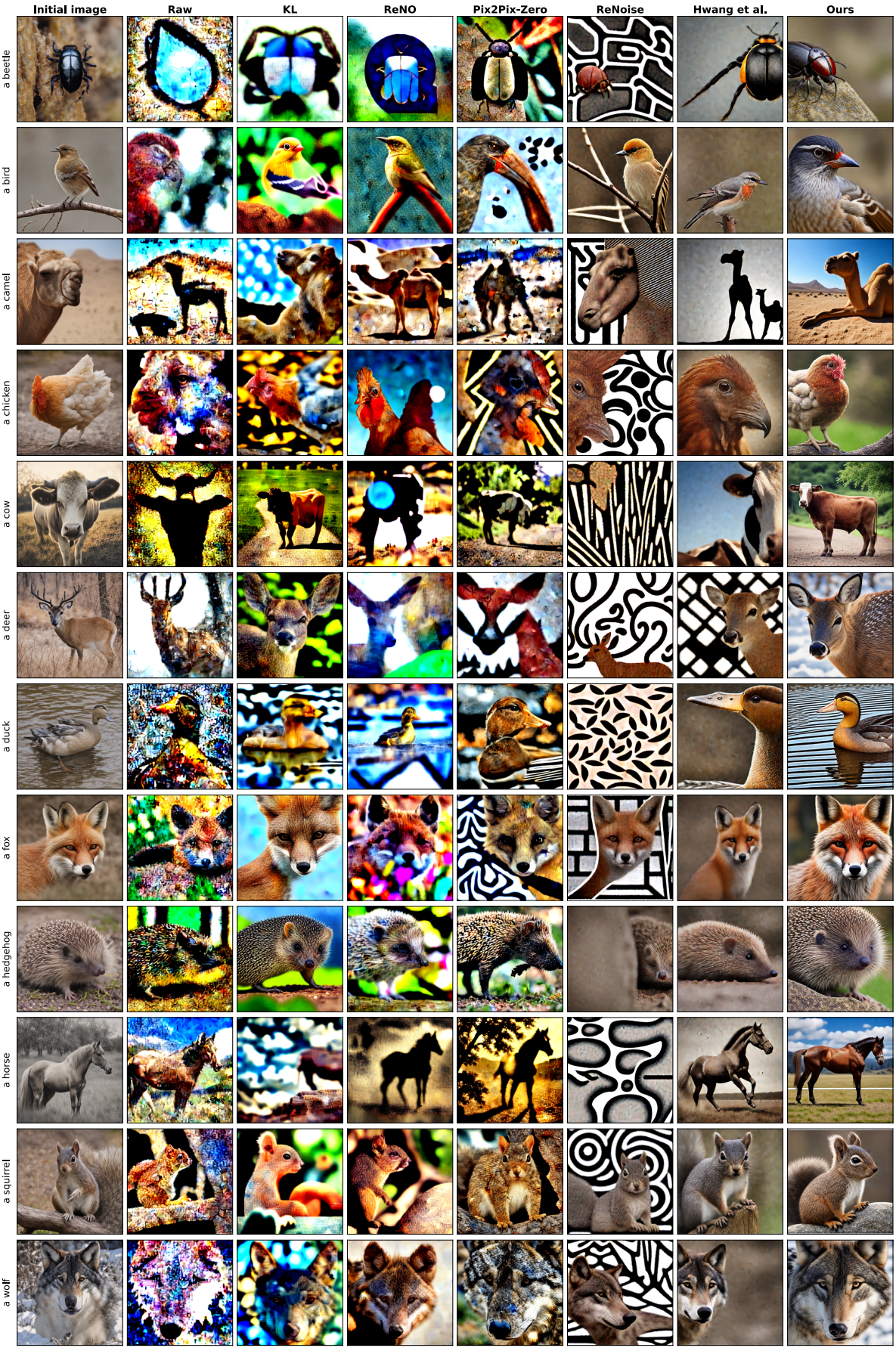

Figure 15. **More Qualitative Results from Aesthetic Image Generation with SD-Turbo.**

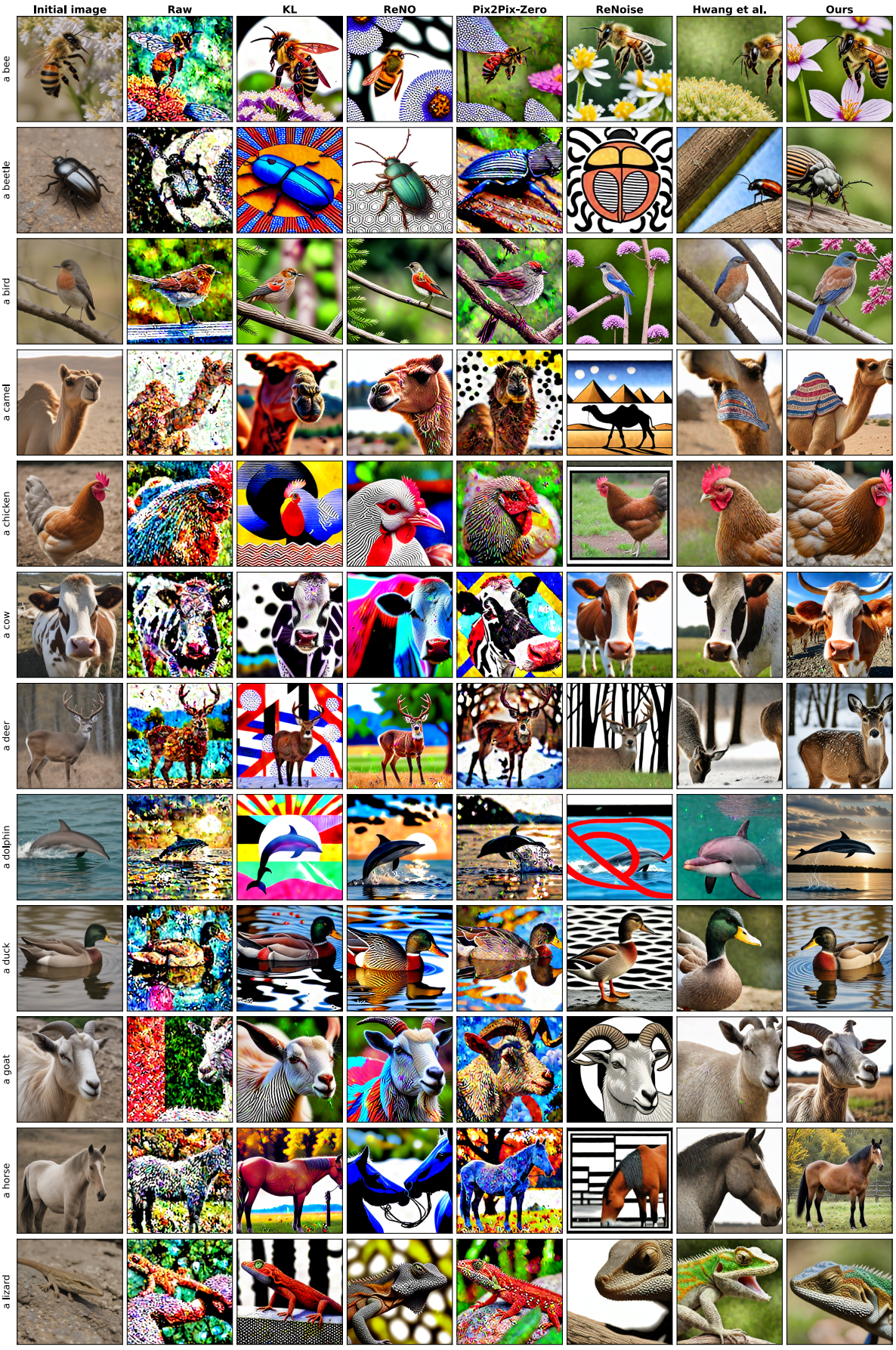

Figure 16. **More Qualitative Results from Aesthetic Image Generation with SDXL-Turbo.**

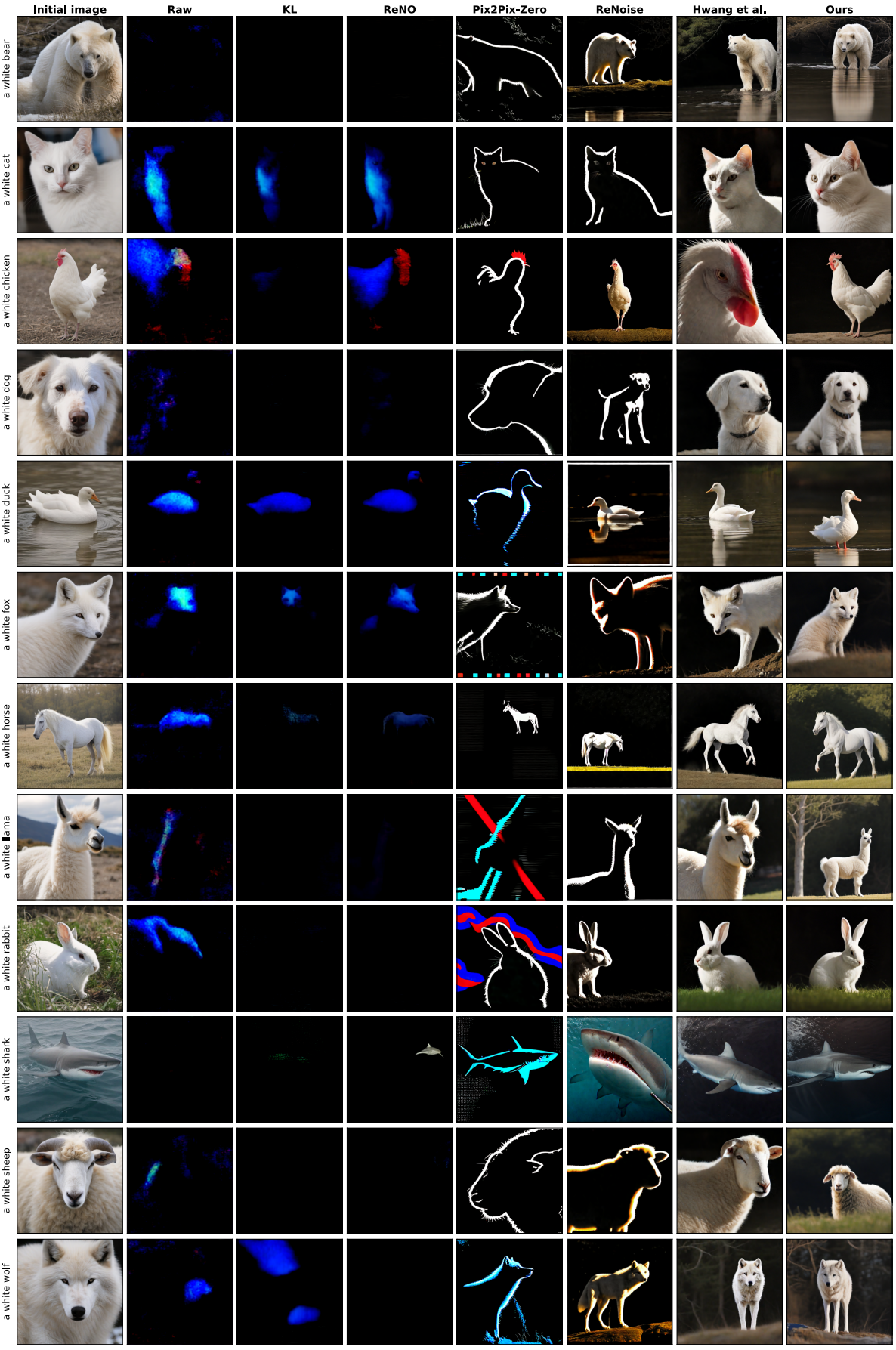

Figure 17. **More Qualitative Results from Brightness Minimization Reward with SDXL-Turbo.**