TRACE ENERGY REGULARIZATION FOR FIDELITY ORIENTED DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models generate high quality synthetic samples, but maintaining fidelity in diffusion models remains challenging, as the differences between real and synthetic data distributions can degrade downstream model performance when synthetic images are used for downstream training. Standard DDPMs are trained to predict Gaussian noise using a mean squared error objective, whose optimal solution is a conditional mean that intends to reduce trace energy (expected squared L^2 norm per dimention of the predicted noise) due to conditional uncertainty. Since the predicted noise directly determines the denoising correction in the reverse process, this shrinkage can produce overly conservative reverse updates. We introduce **EnDiff**, a lightweight regularizer to match the predicted noise trace energy to that of the injected noise, thereby restoring it toward unit trace energy in an isotropic sense. This yields better calibrated reverse updates and promotes high fidelity generation. Across DDPM, Improved Diffusion, and Latent Diffusion Models on CIFAR10/100, CelebA, and CelebA-HQ datasets, EnDiff reveals a consistent improvement in fidelity oriented metrics. Precision increases from 0.590 to 0.648, and Density increases from 1.062 to 1.458 on the CIFAR100 dataset. These results show that trace energy regularization is a simple and effective strategy to shift diffusion models' fidelity diversity balance toward fidelity.

1 INTRODUCTION

Fidelity in data generation captures the degree to which generated data accurately preserve the semantic structure and statistical characteristics of real world data. High fidelity generation has become a critical objective in the modern generative paradigm, given the increasing reliance on synthetic data (Li et al., 2025; Shao et al., 2024) for downstream tasks, where even small imperfections can propagate through learned representations and lead to misinterpretation or degraded performance. This risk is particularly acute in safety critical domains such as healthcare, where imprecise synthetic medical data can lead to false positives and misleading clinical inferences. Furthermore, the importance of high fidelity data generation is particularly evident in recent research across domains such as medical imaging (Wang et al., 2020), speech synthesis (Kong et al., 2020), and remote sensing (Li et al., 2024), where realism and statistical consistency are essential. Within generative modeling frameworks diffusion models (Ho et al., 2020; Sohl-Dickstein et al., 2015) have proven particularly effective for high fidelity data generation, often surpassing alternatives in sample quality, making them a cornerstone for a wide variety of applications.

Diffusion models have emerged as a leading family for high quality generation, including Denoising Diffusion Probabilistic Models (DDPM) (Ho et al., 2020), Improved Diffusion (Nichol & Dhariwal, 2021), and Latent Diffusion Model f(LDM) (Rombach et al., 2022) . They define a forward corruption process that adds Gaussian noise to data to produce noisy inputs x_t, train a network to predict the injected noise across timesteps, and generate samples by reversing this process during inference Ho et al. (2020). Beyond general purpose improvements in quality and efficiency, a growing line of work explicitly targets fidelity in diffusion based generation. These include cascaded diffusion pipelines that emphasize high fidelity synthesis (Ho et al., 2022), diffusion models designed to generate reliable samples from low density regions (Sehwag et al., 2022), physics informed diffusion for high fidelity reconstruction in scientific settings (Shu et al., 2023), and conditional diffusion for structurally consistent medical image synthesis (Dorjsembe et al., 2024). Collectively, these efforts

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



Figure 1: Unconditional samples under an objective only change. DDPM (left) vs. EnDiff (right) on CelebAHQ, CelebA, CIFAR10 and CIFAR100 (rows).

indicate that improving fidelity remains a central and crucial objective in diffusion based generative modeling.

The standard noise prediction Mean Squared Error (MSE) objective does not, in general, enforce the predicted noise to have the same expected squared L^2 norm per dimension as the noise injected by the forward process. We refer to this quantity as trace energy hereafter. The MSE minimizer is a conditional mean predictor whose trace energy can be strictly smaller than the injected noise trace energy. This is shown theoretically in section 3.2 (lemma 1) and shown empirically in fig. 2, where for a baseline DDPM on CIFAR10 and CelebA, the reverse path trace energy drops sharply below 1 near the end of the reverse path.

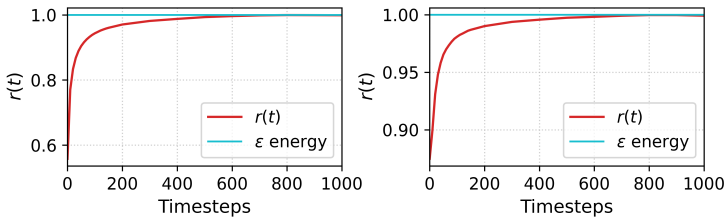
To address this, we propose **EnDiff**, a modification to the training objective that augments the standard DDPM noise prediction loss with a lightweight noise trace energy regularization that encourages the predicted noise trace energy to match the trace energy of the injected noise. The penalty is computed directly from the network output therefore adds negligible overhead, and leaves the diffusion architecture unchanged. Because the intervention is objective only, EnDiff is plug compatible with most diffusion backbones, including DDPMs (Ho et al., 2020), improved diffusion models (Nichol & Dhariwal, 2021), and LDMs (Rombach et al., 2022).

We first study EnDiff in controlled settings using four well defined 2D synthetic datasets such as Swiss Roll, Scattered Moon, Moon with Two Circles, and Central Banana to isolate mechanisms and draw fundamental conclusions about DDPM behavior. Then we validate EnDiff on unconditional image generation across CIFAR10/100 (Krizhevsky et al., 2009), CelebA (Liu et al., 2015), and CelebAHQ (Karras et al., 2018), and across DDPM, improved diffusion and LDM backbones under matched architectures and compute. We report Precision, Recall, Density and Coverage(PRDC) (Sajjadi et al., 2018; Kynkäänniemi et al., 2019; Naeem et al., 2020) alongside FID to characterize fidelity-diversity behavior.

In summary, our contributions are:

- We identify a trace energy mismatch between the predicted noise and the forward process injected noise under the standard diffusion noise prediction MSE objective. We show that the MSE optimal predictor is a conditional mean whose trace energy is strictly smaller than that of the injected noise, which weakens the reverse denoising updates.
- We propose EnDiff, an objective only regularizer that augments the DDPM noise prediction loss with a trace energy matching penalty, biasing the generated distribution toward higher fidelity without changing the architecture, forward process, or sampler.

We support these contributions with (i) controlled studies on four 2D synthetic datasets, (ii) unconditional image generation across DDPM, Improved Diffusion, and LDM backbones on CIFAR10, CIFAR100, CelebA, and CelebAHQ evaluated with FID and PRDC. Representative unconditional samples across the four image datasets are shown in fig. 1.



(a) CIFAR-10: $r(t)$ vs. timestep. (b) CelebA-64: $r(t)$ vs. timestep.

Figure 2: Predicted noise trace energy during reverse sampling (DDPM, CIFAR10 & CelebA). $r(t) = \frac{1}{d}\mathbb{E}_{x_t}\|\epsilon_\theta(x_t, t)\|_2^2$ over reverse steps. The unit trace energy reference $\frac{1}{d}\mathbb{E}\|\epsilon\|_2^2 = 1$ is shown in blue.

2 Related Work

Diffusion models. Diffusion models synthesize samples from Gaussian noise via a learned reverse denoising process (Ho et al., 2020). Improved Diffusion models that refine training and parameterization choices (Nichol & Dhariwal, 2021), and LDMs that enable scalable high resolution synthesis by operating in a learned latent space (Rombach et al., 2022). Furthermore, high resolution fidelity can be improved through cascaded diffusion pipelines (Ho et al., 2022).

Diffusion model Improvements. A broad set of methods improves diffusion models by intervening at different stages of the pipeline including objective weighting, forward-process design, or sampling modifications. Min-SNR reweights training across timesteps (Hang et al., 2023). Other approaches redesign the forward corruption process, for example by replacing white Gaussian noise with structured alternatives (blue-noise) (Huang et al., 2024) or using edge-preserving noise schedules to retain boundaries and fine details during corruption (Vandersanden et al., 2026). Complementary sampling time methods accelerate inference by modifying or compressing the denoising trajectory while reusing the same trained denoiser, such as DDIM (Song et al., 2021) and progressive distillation (Salimans & Ho, 2022). EnDiff instead regularizes a scalar second order statistic of the predicted noise while leaving the forward process, architecture, and sampler unchanged.

3 Methods

3.1 Background

DDPM defines a forward corruption process that gradually adds Gaussian noise using a variance schedule $\beta_t \in (0, 1)$ over timesteps $t = 1, \ldots, T$. Define $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. The forward process admits a closed form reparameterization that produces training pairs without simulating the full chain using,

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1 - \bar{\alpha}_t}\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \mathbf{I}). \tag{1}$$

Here $x_0 \sim p_{\text{data}}$ and ϵ is the injected noise. As t increases, x_t approaches the Gaussian prior $\mathcal{N}(0, \mathbf{I})$. The predictor ϵ_θ is trained to predict the injected noise by minimizing the standard noise objective,

$$L_{\text{simple}} = \mathbb{E}_{t, x_0, \epsilon}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|_2^2\right]. \tag{2}$$

We sample $x_T \sim \mathcal{N}(0, \mathbf{I})$ and for $t = T, \ldots, 1$ apply the reverse transition with fixed posterior variance. Under noise-prediction, reverse mean is

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}}\, \epsilon_\theta(x_t, t)\right). \tag{3}$$

3.2 Trace Energy Shrinkage Under Noise Prediction

DDPM training predicts the injected noise ϵ in the reparameterization $x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1 - \bar{\alpha}_t}\, \epsilon$, where $\epsilon \sim \mathcal{N}(0, \boldsymbol{I})$.

Let $U \triangleq (x_t, t)$ denote the conditioning variable provided to a predictor $f(U)$. Here f denotes a generic predictor; the analysis below holds for any measurable f, and f specializes to the trained network $\epsilon_\theta(x_t, t)$ in section 3.3.

The noise MSE objective is

$$\mathcal{L}(f) \triangleq \mathbb{E}_{t,x_0,\epsilon}\big[\|\epsilon - f(U)\|_2^2\big], \tag{4}$$

where $t \sim \mathrm{Unif}\{1, \ldots, T\}$, $x_0 \sim p_{\mathrm{data}}$, and $\epsilon \sim \mathcal{N}(0,)$, matching the DDPM training procedure. Define the conditional mean

$$m(U) \triangleq \mathbb{E}[\epsilon \mid U]. \tag{5}$$

Lemma 1. *Assume $\mathbb{E}\big[\|\epsilon\|_2^2\big] < \infty$, and let $f(U)$ be any measurable predictor satisfying $\mathbb{E}\big[\|f(U)\|_2^2\big] < \infty$,*

$$\mathbb{E}\big[\|\epsilon - f(U)\|_2^2\big] = \mathbb{E}\big[\mathrm{Tr}\big(\mathrm{Cov}(\epsilon \mid U)\big)\big] + \mathbb{E}\big[\|m(U) - f(U)\|_2^2\big]. \tag{6}$$

Equation (6) has two relevant consequences. First, since the first term on the right hand side is independent of f, $\mathcal{L}(f)$ is minimized uniquely by the conditional mean $m(U)$

$$f^\star(U) = m(U). \tag{7}$$

Thus, the optimal predictor under the MSE objective is the conditional mean of the injected noise given U.

Second, evaluating eq. (6) at the trivial predictor $f(U) \equiv 0$ gives the conditional second-moment decomposition,

$$\mathbb{E}\|\epsilon\|_2^2 = \mathbb{E}\|m(U)\|_2^2 + \mathbb{E}\big[\mathrm{Tr}\big(\mathrm{Cov}(\epsilon \mid U)\big)\big]\,. \tag{8}$$

Because $\epsilon \sim \mathcal{N}(0, \boldsymbol{I}_d)$ implies $\mathbb{E}\|\epsilon\|_2^2 = d$, define the expected squared L^2 norm per dimension of the MSE-optimal predictor, which we refer to as its trace energy as r,

$$r \triangleq \frac{1}{d}\mathbb{E}_U\big[\|f^\star(U)\|_2^2\big] = \frac{1}{d}\mathbb{E}_U\big[\|m(U)\|_2^2\big]. \tag{9}$$

Using eq. (8) gives

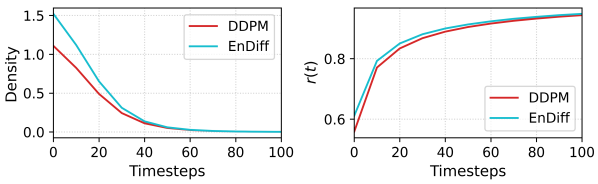
$$r = 1 - \frac{1}{d}\mathbb{E}\big[\mathrm{Tr}\big(\mathrm{Cov}(\epsilon \mid U)\big)\big] \le 1, \tag{10}$$

with strict inequality whenever $\mathrm{Tr}(\mathrm{Cov}(\epsilon \mid U)) > 0$ with nonzero probability. In diffusion, ϵ is generally not uniquely determined by $U = (x_t, t)$ because x_t mixes the unknown x_0 and ϵ whenever $0 < \bar{\alpha}_t < 1$. Therefore, the conditional distribution, $\epsilon \mid U$ is typically non-degenerate and $r < 1$ is expected. Thus, sub-unit trace energy ($r < 1$) is a consequence of conditional uncertainty under MSE prediction, with the gap exactly equal to the normalized expected conditional covariance trace. The proof of lemma 1 is given in Appendix section A.1.

3.3 Energy Regularized Noise Prediction

Section 3.2 identifies a systematic difference between the second moment of the MSE optimal conditional mean and that of the injected Gaussian. We augment DDPM training with an energy regularization term that encourages the trace energy of $\epsilon_\theta(x_t, t)$ to match the unit trace energy of the injected noise $\epsilon \sim \mathcal{N}(0, \boldsymbol{I}_d)$. This regularizer biases the learned predictor away from the pure MSE-optimal conditional mean so that its trace energy more closely matches that of the injected noise. We formulate the following constrained optimization problem:

$$\min_\theta \quad \mathcal{L}_{\mathrm{simple}}(\theta) \quad \text{subject to} \quad \mathbb{E}_{t,x_0,\epsilon}\left[\frac{1}{d}\,\|\epsilon_\theta(x_t, t)\|_2^2\right] = 1. \tag{11}$$



(a) CIFAR-10: Density vs. timestep (0–100). (b) CIFAR-10: $r(t)$ vs. timestep (0–100).

Figure 3: CIFAR10 (timesteps 0-100): Density and predicted noise trace energy. Left: Density vs. diffusion timestep t for DDPM and EnDiff. Right: predicted noise trace energy $r(t)$ vs. t. EnDiff yields higher Density and higher $r(t)$ in this regime, consistent with reduced predicted-noise trace-energy underestimation during late denoising.

We define the trace energy residual as

$$g(\theta) \triangleq 1 - \mathbb{E}_{t,x_0,\epsilon}\left[\frac{1}{d}\left\|\epsilon_\theta(x_t,t)\right\|_2^2\right], \tag{12}$$

where the expectation is taken under the same sampling procedure as in $\mathcal{L}_{\text{simple}}$. We then optimize the penalized objective

$$\mathcal{L}_{\text{EnDiff}}(\theta) \triangleq \mathcal{L}_{\text{simple}}(\theta) + \lambda\big(g(\theta)\big)^2, \qquad \lambda > 0. \tag{13}$$

The normalization by d keeps the penalty scale comparable across dimensionalities. Importantly, this regularizer matches a necessary, but not sufficient, second-order statistic for isotropy. Specifically, if a random vector Z satisfies $\mathbb{E}[Z] = 0$ and $\mathrm{Cov}(Z) = \boldsymbol{I}_d$, then $\mathbb{E}[\|Z\|_2^2/d] = 1$. Enforcing $g(\theta) \approx 0$ therefore regularizes the average magnitude of $\epsilon_\theta(x_t, t)$ without explicitly constraining its full covariance structure. The training loss uses the penalty $\lambda\big(g(\theta)\big)^2$. This adds negligible computational overhead, since the squared norm is computed from the predictor output already produced for the MSE term.

3.4 Effect of Energy Regularization

The trace energy mismatch identified in section 3.2 directly affects reverse sampling. From eq. (3), the predicted noise $\epsilon_\theta(x_t, t)$ determines the noise dependent correction applied to x_t when computing the reverse mean. If its magnitude is systematically underestimated, this correction becomes smaller and the reverse transition becomes overly conservative. Accumulated over successive reverse steps, such under correction can limit the refinement of samples toward well supported regions of the data distribution, particularly during late denoising where local structure is progressively recovered.

EnDiff mitigates this effect by calibrating the average magnitude of the predicted noise while leaving its sample dependent direction primarily determined by the standard denoising objective. Since the regularizer constrains only an aggregate second order statistic rather than the full covariance or per sample structure of $\epsilon_\theta(x_t, t)$, it preserves substantial model flexibility. We hypothesize that this better calibrated denoising field helps preserve structural distinctions in the learned data distribution and reduces the tendency of reverse trajectories to remain in weakly supported or intermediate regions. Consequently, sampling becomes more concentrated around high density regions of the true data distribution, leading to a fidelity oriented shift in generation.

We examine this mechanism empirically by comparing the predicted-noise trace energy and generation Density along the reverse trajectory. Figure 3(b) indicates that, toward the end of the reverse trajectory, standard diffusion tends to produce systematically smaller values of $r(t)$, while EnDiff keeps this statistic higher than DDPM. The corresponding generation behavior is shown in Figure 3(a), where EnDiff consistently achieves higher Density than DDPM during late denoising. The coincidence between reduced trace energy underestimation and increased Density supports the proposed mechanism: better calibrated predicted noise magnitudes mitigate overly conservative reverse updates and progressively concentrate generated samples in regions with stronger support under the true data distribution. Additional transient curves of other datasets are provided in Appendix section C.1.

Algorithm 1 Training EnDiff (trace energy matching)

Require: dataset $\mathcal{D}$, schedule $\{\beta_t\}_{t=1}^{T}$ and $\bar{\alpha}_t$, weight λ, batch size B

1: **repeat**
2: sample $\{x_0^{(i)}\}_{i=1}^{B} \sim \mathcal{D}$, $\{t^{(i)}\}_{i=1}^{B} \sim \mathrm{Unif}\{1,\dots,T\}$, and $\{\epsilon^{(i)}\}_{i=1}^{B} \sim \mathcal{N}(0, \boldsymbol{I})$
3: $x_t^{(i)} \leftarrow \sqrt{\bar{\alpha}_{t^{(i)}}}\, x_0^{(i)} + \sqrt{1-\bar{\alpha}_{t^{(i)}}}\, \epsilon^{(i)}$ for $i = 1,\dots,B$
4: $\hat{\epsilon}^{(i)} \leftarrow \epsilon_\theta(x_t^{(i)}, t^{(i)})$ for $i = 1,\dots,B$
5: $\mathcal{L} \leftarrow \frac{1}{B}\sum_{i=1}^{B} \left\|\epsilon^{(i)} - \hat{\epsilon}^{(i)}\right\|_2^2 + \lambda \left(\frac{1}{B}\sum_{i=1}^{B} \frac{1}{d}\left\|\hat{\epsilon}^{(i)}\right\|_2^2 - 1\right)^2$
6: take a gradient step on $\nabla_\theta \mathcal{L}$
7: **until** converged

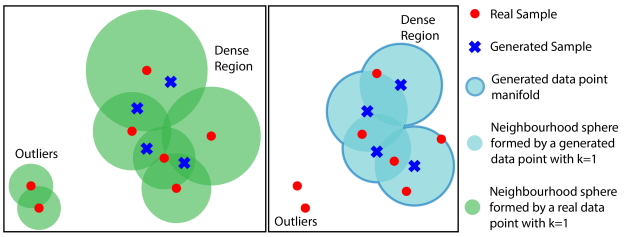


Figure 4: An example scenario for illustrating a situation where high Density and low Coverage is recorded. Generating samples in the neighborhoods of the highly dense regions over the outliers in the true manifold has resulted in a high Density and low Coverage.

Furthermore, algorithm 1 summarizes the EnDiff training procedure. The only modification to standard DDPM training is the addition of a lightweight energy penalty computed from the predicted noise while the forward noising process, timestep sampling strategy, and network architecture remain unchanged. As a result, EnDiff is directly compatible with common DDPM improvements and can be combined with most sampling and training variants without requiring method specific changes.

3.5 Interpretation of Evaluation Metrics

Since single number scores can hide quality coverage trade offs, Precision, Recall style diagnostics were proposed to disentangle sample quality from support coverage (Sajjadi et al., 2018; Kynkäänniemi et al., 2019). PRDC extends this view with Density and Coverage to better diagnose local realism versus global support (Naeem et al., 2020). We adopt these metrics to quantify the fidelity-diversity shifts induced by our objective only modification, and report them alongside FID as a standard perceptual distribution distance baseline (Heusel et al., 2017).

Fréchet Inception Distance (FID). Measures the similarity between real and generated distributions in the feature space of a pretrained Inception network. Lower FID indicates greater similarity between their feature statistics and global structure (Heusel et al., 2017).

Precision, Recall, Density, and Coverage. A neighborhood sphere is the region around a point with radius equal to the distance to its k-th nearest neighbor. The collections of spheres around real and generated samples define the real and generated manifolds, respectively (fig. 4).

Precision is the fraction of generated samples lying within the real manifold and reflects generated-sample fidelity (Sajjadi et al., 2018). In the left of fig. 4, all four generated samples lie within the real manifold; therefore, Precision is $4/4$.

Recall is the fraction of real samples lying within the generated manifold and reflects support coverage (Sajjadi et al., 2018; Kynkäänniemi et al., 2019). However, the lack of generated samples near sparse outliers in the real data leads to low Recall, as the generated manifold fails to capture these regions. In the right of fig. 4, five of the seven real samples are covered; therefore, Recall is $5/7$.

Table 1: PRDC on 2D synthetic datasets for baseline DDPM vs. EnDiff$_{ddpm}$. Higher is better (↑).

Dataset	Model	Prec.↑	Rec.↑	Den.↑	Cov.↑
Central Banana	DDPM	0.9740	**0.9983**	0.9673	**0.9629**
Central Banana	EnDiff$_{ddpm}$	**0.9842**	0.9966	**0.9759**	0.9534
Moon Circles	DDPM	0.9914	**0.9982**	0.9930	**0.9637**
Moon Circles	EnDiff$_{ddpm}$	**0.9954**	0.9965	**1.0047**	0.9609
Scattered Moon	DDPM	0.9991	0.9957	1.0061	0.9640
Scattered Moon	EnDiff$_{ddpm}$	**0.9996**	**0.9960**	**1.0062**	**0.9664**
Swiss Roll	DDPM	0.9829	**0.9976**	0.9783	**0.9527**
Swiss Roll	EnDiff$_{ddpm}$	**0.9872**	0.9943	**0.9865**	0.9472

Density counts the real-sample neighborhood spheres that contain each generated sample, rewarding samples located in densely populated real regions. This captures the local context of a distribution by measuring how closely generated samples lie to densely packed real samples (Naeem et al., 2020). In the left of fig. 4, the overlap counts are 3, 2, 2, and 1; therefore, Density is $8/7$.

Coverage is the fraction of real samples whose neighborhood contains at least one generated sample and reflects the represented diversity of the distribution. However, sparse outliers in the real manifold may lead to low Coverage when no generated samples occur near those regions (Naeem et al., 2020). In the left of fig. 4, five of the seven real neighborhoods contain a generated sample; therefore, Coverage is $5/7$.

4 Experiments

4.1 Experimental Setup

2D Synthetic Data. To isolate the effect of the proposed objective under controlled geometry, we evaluate DDPM and EnDiff on four 2D synthetic distributions: Swiss Roll, Scattered Moon, Moon with Two Circles, and Central Banana. Details of these synthetic datasets construction are provided in Appendix section B. We use a conditional dense network with three fully connected hidden layers and ReLU activations, a learning rate of 10^{-3}, a linear variance schedule, 1000 diffusion timesteps, and 1000 training epochs. For each dataset, the forward process, network, optimizer, and training budget are identical, and only the objective is modified by adding the energy penalty in eq. (13). Unless stated otherwise, sampling uses the standard 1000-step ancestral reverse update. We report Precision, Recall, Density, and Coverage directly in $\mathbb{R}^2$. Full PRDC settings and alternative variance schedules are provided in Appendix sections C.4 and D.2.

Image Generation. We evaluate EnDiff as an objective-only modification across DDPM, Improved Diffusion (Imp), and latent diffusion (LDM). For each baseline and its corresponding EnDiff variant, we keep the architecture, data pipeline, optimizer, and training budget identical and change only the training loss by adding the penalty in eq. (13). We evaluate DDPM and EnDiff$_{ddpm}$, Imp and EnDiff$_{imp}$, and LDM and EnDiff$_{ldm}$ on CIFAR10 and CIFAR100 (32×32), CelebA (64×64), and CelebAHQ (256×256) for unconditional generation. Sampling uses the standard 1000-step ancestral reverse update. We compute FID and PRDC using 50K generated samples for CIFAR10, CIFAR100, and CelebA, and 27K samples for CelebAHQ. Unless stated otherwise, we use $\lambda = 0.3$ across all datasets and model variants to keep the regularizer secondary to the denoising objective, while still providing a consistent energy penalty during training. We keep λ fixed across datasets and model variants to reduce hyperparameter degrees of freedom and isolate the effect of the proposed objective. A sensitivity analysis for λ and the full training, sampling, and evaluation configurations are provided in Appendix section D.1, section C.2, section C.3 and section C.4.

4.2 2D Synthetic Data Generation

We believe that the disparity in the changes of Precision, Recall, Density, and Coverage is a direct consequence of imposing an trace energy constraint on the noise prediction objective. By discour-

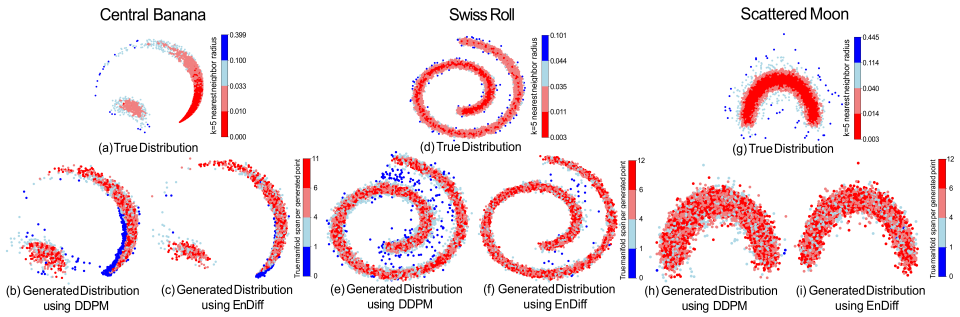


Figure 5: Central Banana, Swiss Roll, and Scattered Moon 2D synthetic datasets: (a),(d),(g) True distribution points, color coded by k=5 nearest neighbor radius. (b),(e),(h) DDPM-generated points, color-coded by true manifold span per point. (c),(f),(i) EnDiff generated points, color coded by true manifold span per point.

aging underestimated predicted-noise trace energy (section 3.3), EnDiff produces less conservative reverse updates and encourage samples to concentrate near dense regions of the true distribution. Under PRDC, this generally increases fidelity-oriented metrics (Precision/Density) while reducing diversity oriented metrics (Recall/Coverage).

The Central Banana distribution contains a detached mode and a discontinuity in its density structure (fig. 5(a)). EnDiff better preserves this discontinuity and produces fewer samples that incorrectly interpolate between the main lobe and the detached mode (fig. 5(b,c)). It also generates fewer samples in the low-density tail of the main lobe, consistent with the increased Precision and Density in table 1. A similar pattern appears for Swiss Roll (fig. 5(d-f)). The DDPM samples drift away from the intended rolled structure, whereas EnDiff better follows the true manifold. The corresponding increase in Density and decrease in Coverage for Swiss Roll and Central Banana are consistent with these observations (table 1). For Scattered Moon, the true distribution includes scattered low-probability points around its main structure (fig. 5(g)). EnDiff captures the prominent structure without being strongly influenced by these low probability scattered regions, whereas DDPM shows limitations in distinguishing the main structure from the scattered noise (fig. 5(h,i)).

This helps explain why the percentage improvements differ across synthetic datasets (table 1). In particular, for Scattered Moon, which contains a higher proportion of low probability scattered regions, the fidelity gains are comparatively marginal. Since each PRDC metric has utility depending on the application (Kynkäänniemi et al., 2019), these results further motivate evaluating diffusion models beyond FID by explicitly separating local fidelity (Precision/Density) from support coverage (Recall/Coverage).

4.3 Image Generation

Table 2 summarizes unconditional generation across four datasets and three diffusion backbones under an objective only modification. Across model families, we observe a consistent shift in PRDC. EnDiff increases the fidelity oriented metrics Precision and Density, while Recall often decreases, reflecting a fidelity diversity trade off under this evaluation. Coverage remains high in most settings, suggesting that the Density gains are not accompanied by a large loss of support over the data manifold, but rather by improved local sample quality as measured by PRDC. The same fidelity oriented pattern holds under class conditional generation on CIFAR10, where Precision, Density, and Coverage improve with FID nearly unchanged; results are given in Appendix section C.5.

For pixel-space DDPMs, this pattern is already present, with consistent Precision and Density gains across datasets and mild Recall reductions on CIFAR10, CIFAR100, and CelebA. For example, on CIFAR100, Density increases from 1.062 to 1.458 while Coverage increases from 0.965 to 0.987, which is consistent with improved local realism under PRDC without an accompanying drop in coverage. FID does not improve uniformly under this trade off, however, CelebAHQ exhibits an aligned improvement where FID decreases alongside PRDC fidelity gains. Qualitatively, the EnDiff

Table 2: Unconditional generation across diffusion backbones under matched architectures and training budgets. EnDiff denotes the corresponding backbone-specific variant.

		CIFAR10		CIFAR100		CelebA		CelebAHQ	
Diffusion Backbone	Metric	Base	EnDiff	Base	EnDiff	Base	EnDiff	Base	EnDiff
DDPM	FID ↓	**3.443**	3.645	**4.966**	5.503	**3.574**	4.148	14.7246	**14.188**
	Prec. ↑	0.606	**0.661**	0.590	**0.648**	0.7301	**0.759**	0.765	**0.780**
	Recall ↑	**0.532**	0.485	**0.523**	0.471	**0.624**	0.603	0.378	**0.406**
	Dens. ↑	1.090	**1.427**	1.062	**1.458**	1.133	**1.274**	1.951	**2.114**
	Cov. ↑	0.970	**0.985**	0.965	**0.987**	0.969	**0.978**	0.966	**0.982**
Improved Diff.	FID ↓	**4.072**	10.060	**6.856**	13.376	**4.196**	7.799	**10.694**	12.956
	Prec. ↑	0.597	**0.637**	0.501	**0.560**	0.696	**0.725**	0.676	**0.712**
	Recall ↑	**0.558**	0.416	**0.485**	0.452	**0.539**	0.524	**0.397**	0.375
	Dens. ↑	0.999	**1.702**	0.925	**1.196**	1.368	**1.552**	1.821	**2.032**
	Cov. ↑	0.962	**0.992**	0.947	**0.972**	0.984	**0.990**	0.989	**0.990**
LDM	FID ↓	**10.361**	10.447	**14.591**	15.597	**7.557**	7.909	**10.234**	10.403
	Prec. ↑	0.674	**0.712**	0.662	**0.680**	0.773	**0.776**	0.709	**0.717**
	Recall ↑	**0.396**	0.370	**0.399**	0.380	**0.476**	**0.476**	**0.417**	0.410
	Dens. ↑	1.726	**2.013**	1.729	**1.900**	1.741	**1.766**	2.206	**2.328**
	Cov. ↑	0.992	**0.995**	0.989	**0.992**	0.992	**0.993**	0.999	**1.000**

samples in fig. 1 show cleaner local features, consistent with the observed improvements in Precision and Density. Additional qualitative results per dataset are provided in the Appendix section E.

Improved Diffusion follows the same qualitative PRDC shift, with Precision and Density increasing while Recall decreases. In this architecture, FID can degrade despite PRDC fidelity improvements. On CIFAR10, Density increases from 0.999 to 1.702 while Recall decreases from 0.558 to 0.416. This kind of mismatch between likelihood oriented training choices and FID has also been reported for Improved Diffusion, where objective changes can improve likelihood related behavior while worsening FID (Nichol & Dhariwal, 2021). We therefore treat PRDC and FID as complementary signals rather than expecting monotonic agreement.

EnDiff remains beneficial for latent diffusion, with smaller but consistent gains in Precision and Density and minimal changes in Coverage. The reduced magnitude is consistent with latent space modeling already imposing strong representational constraints, nevertheless, the directionally consistent PRDC changes across DDPM, Improved Diffusion, and LDM suggest that EnDiff acts as a backbone agnostic objective level bias toward fidelity oriented generation.

5 Conclusion

We introduced **EnDiff**, an objective only regularizer for DDPM style noise prediction that penalizes deviations of the trace energy of $\epsilon_\theta(x_t, t)$ from the injected noise of the forward process. Our analysis shows that the optimal predictor under MSE is a conditional mean whose expected trace energy can be strictly smaller than of the injected noise, potentially weakening reverse denoising updates. Across DDPM, Improved Diffusion, and latent diffusion backbones on CIFAR10/100, CelebA, and CelebAHQ, EnDiff shifts PRDC toward higher fidelity oriented scores, particularly Precision and Density, while Recall decreasing in several settings. These results demonstrate that EnDiff provides a lightweight objective-level mechanism for biasing diffusion models toward fidelity without modifying the architecture, forward process, or sampler. This work also highlights the benefits of utilizing evaluation metrics beyond FID, as Precision, Recall, Density, and Coverage provide additional insight into fidelity and diversity. We firmly believe that our work offers other researchers novel perspectives to explore.

AI USE STATEMENT

In this work, we used generative AI tools for the related work section and to help improve grammar, clarity, and readability of the text drafted by the authors. We did not use generative AI tools for research tasks, including the development of theoretical models or conceptual frameworks, formulation of mathematical claims, writing of mathematical proofs, refinement of hypotheses, design of research methodology, implementation of methods. Other required disclosure tasks, including synthetic dataset generation, translation, dataset cleaning or reformatting, and qualitative or thematic data analysis, were not applicable to this work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

5.1 ETHICS STATEMENT

We do not have any concerns.

REFERENCES

Zolnamar Dorjsembe, Hsing-Kuo Pao, Sodtavilan Odonchimed, and Furen Xiao. Conditional diffusion models for semantic 3D brain MRI synthesis. *IEEE Journal of Biomedical and Health Informatics*, 28(7):4084–4093, 2024. doi: 10.1109/JBHI.2024.3385504.

Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via Min-SNR weighting strategy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7407–7417, 2023. doi: 10.1109/ICCV51070.2023.00684.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6626–6637, 2017.

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

Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high-fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022. URL https://jmlr.org/papers/v23/21-0635.html.

Xingchang Huang, Corentin Salaün, Cristina Vasconcelos, Christian Theobalt, A. Cengiz Öztireli, and Gurprit Singh. Blue noise for diffusion models. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11, 2024. doi: 10.1145/3641519.3657435.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Hk99zCeAb.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. HiFi-GAN: Generative adversarial networks for efficient and high-fidelity speech synthesis. In *Advances in Neural Information Processing Systems*, volume 33, pp. 17022–17033, 2020.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. CIFAR-10 (Canadian Institute for Advanced Research), 2009. URL https://www.cs.toronto.edu/~kriz/cifar.html.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Ao Li, Minchao Wu, Rui Ouyang, Yongming Wang, Fan Li, and Zhao Lv. A multimodal-driven fusion data augmentation framework for emotion recognition. *IEEE Transactions on Artificial Intelligence*, 6(8):2083–2097, 2025. doi: 10.1109/TAI.2025.3537965.

Yue Li, Xiaorui Wang, Chao Zhang, Zhonggen Zhang, and Fafa Ren. High-fidelity infrared remote sensing image generation method coupled with the global radiation scattering mechanism and Pix2PixGAN. *Remote Sensing*, 16(23):4350, 2024. doi: 10.3390/rs16234350.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3730–3738, 2015. doi: 10.1109/ICCV.2015.425.

Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7176–7185, 2020.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8162–8171, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022. doi: 10.1109/CVPR52688.2022.01042.

Mehdi S. M. Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. In *Advances in Neural Information Processing Systems*, volume 31, pp. 5234–5243, 2018.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI.

Vikash Sehwag, Caner Hazirbas, Albert Gordo, Firat Ozgenel, and Cristian Canton Ferrer. Generating high-fidelity data from low-density regions using diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11482–11491, 2022. doi: 10.1109/CVPR52688.2022.01120.

Zhuchen Shao, Liuxi Dai, Yifeng Wang, Haoqian Wang, and Yongbing Zhang. AugDiff: Diffusion-based feature augmentation for multiple instance learning in whole slide image. *IEEE Transactions on Artificial Intelligence*, 5(12):6617–6628, 2024. doi: 10.1109/TAI.2024.3454591.

Dule Shu, Zijie Li, and Amir Barati Farimani. A physics-informed diffusion model for high-fidelity flow field reconstruction. *Journal of Computational Physics*, 478:111972, 2023. doi: 10.1016/j.jcp.2023.111972.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265, 2015.

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

Jente Vandersanden, Sascha Holl, Xingchang Huang, and Gurprit Singh. Edge-preserving noise for diffusion models. *Computer Graphics Forum*, 45(2):e70383, 2026. doi: 10.1111/cgf.70383.

Clinton J. Wang, Natalia S. Rost, and Polina Golland. Spatial-intensity transform GANs for high-fidelity medical image-to-image translation. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*, volume 12262 of *Lecture Notes in Computer Science*, pp. 749–759, Cham, 2020. Springer. doi: 10.1007/978-3-030-59713-9_72.

A Proofs and Additional Derivations

A.1 Proof of Lemma 1

Proof. Let $m(U) = \mathbb{E}[\epsilon \mid U]$. Decompose

$$\epsilon - f(U) = (\epsilon - m(U)) + (m(U) - f(U)).$$

Expanding the squared norm yields

$$\|\epsilon - f(U)\|_2^2 = \|\epsilon - m(U)\|_2^2 + \|m(U) - f(U)\|_2^2 + 2\langle \epsilon - m(U),\, m(U) - f(U)\rangle.$$

Taking conditional expectation given U, the cross term vanishes because $\mathbb{E}[\epsilon - m(U) \mid U] = \mathbb{E}[\epsilon \mid U] - m(U) = 0$, hence

$$\mathbb{E}\big[\|\epsilon - f(U)\|_2^2 \mid U\big] = \mathbb{E}\big[\|\epsilon - m(U)\|_2^2 \mid U\big] + \|m(U) - f(U)\|_2^2.$$

By definition,

$$\mathrm{Cov}(\epsilon \mid U) = \mathbb{E}\big[(\epsilon - m(U))(\epsilon - m(U))^\top \mid U\big].$$

Taking trace and using $\mathrm{Tr}(vv^\top) = \|v\|_2^2$ for vectors v gives

$$\mathrm{Tr}\left(\mathrm{Cov}(\epsilon \mid U)\right) = \mathbb{E}\big[\|\epsilon - m(U)\|_2^2 \mid U\big].$$

Substituting into the previous display and taking expectation over U yields eq. (6).

B Synthetic Dataset Construction

We use four procedurally generated two-dimensional datasets to evaluate diffusion models under controlled geometric structures. Each dataset contains 10,000 samples generated with random seed 42 and is independently normalized to the range $[-1, 1]$ along each coordinate.

Scattered Moon. The dataset consists of a crescent-shaped manifold containing 9,500 samples and 500 scattered samples obtained by adding Gaussian perturbations ($\sigma = 0.3$) to randomly selected crescent points. The crescent itself is generated with Gaussian noise of standard deviation 0.1.

Swiss Roll. The Swiss Roll is obtained by projecting a three-dimensional Swiss-roll manifold onto two dimensions. Gaussian noise with standard deviation 0.1 is added during generation, producing a curved spiral-shaped distribution.

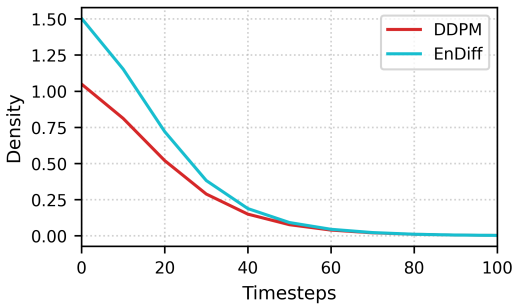
Central Banana. The dataset contains a banana-shaped tail and a dense central cluster. The tail is generated from an exponentially decaying curved trajectory with uniformly distributed local perturbations, while the central cluster consists of 500 Gaussian-distributed samples positioned near the middle of the manifold. This construction produces a gradual transition from a dense region to a lower-density curved tail.

Moon with Two Circles. The dataset combines a crescent-shaped manifold with two compact circular clusters centred at $(-1, -0.75)$ and $(1, -0.75)$. Each circle contains 500 samples with nominal radius 0.2, while the remaining samples form the crescent. Small Gaussian perturbations are applied to all components.

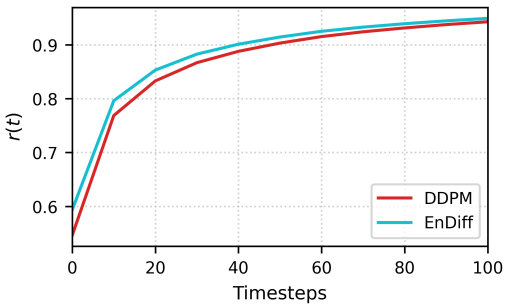
C Transient Sampling Curves and Sensitivity Analysis

C.1 Transient Sampling Curves

Figure 6 shows the Density and predicted-noise trace energy for CIFAR100 over the low-noise timestep range. Figures 7 and 8 present the PRDC metrics and predicted-noise trace energy for CIFAR10 and CIFAR100 over the full reverse trajectory and the corresponding low-noise timestep range.



(a) CIFAR-100: Density vs. timestep (0–100).



(b) CIFAR-100: $r(t)$ vs. timestep (0–100).

Figure 6: CIFAR100 (timesteps 0-100): Density and predicted-noise trace energy. Left: PRDC Density vs. diffusion timestep t for DDPM and EnDiff. Right: predicted noise trace energy diagnostic $r(t)$ vs. t. EnDiff yields higher Density and higher $r(t)$ in this regime, consistent with reduced predicted-noise trace-energy underestimation during late denoising.

C.2 Training Configurations

This appendix provides a modular description of all training configurations used in our experiments. Unless otherwise stated, all models were trained using the Adam optimizer without weight decay. We used a diffusion process with 1000 timesteps and applied an exponential moving average (EMA) to model parameters, using decay factors in the range 0.9999. Sampling used the standard ancestral reverse diffusion update. Gradient clipping was enabled for datasets with a maximum norm of 1.0. Dropout used was 0.1 for DDPM and 0.3 for Improved Diffusion. These global defaults form the baseline for the more detailed specifications below.

C.2.1 Pixel-Space DDPMs.

All DDPMs follow the design outlined in the original DDPM Ho et al. (2020) and share a common architectural structure unless a dataset specific override is noted.

Architecture We used a ResNet based U-Net with sinusoidal time embeddings. Attention layers were inserted at an intermediate spatial resolution (e.g., 16×16 for CIFAR and CelebA). Each resolution level contains two residual blocks. Channel multipliers vary by dataset:

- CIFAR10/CIFAR100: $[1, 2, 2, 2]$ with base width 128.
- CelebA (64×64): $[1, 2, 2, 2, 4]$ with base width 128.
- CelebAHQ (256×256): $[1, 1, 2, 2, 4, 4]$ with base width 128.

Diffusion Setup All DDPM models used the linear variance schedule with $\beta_1 = 10^{-4}$ and $\beta_T = 2 \times 10^{-2}$. The U-Net predicts $\epsilon_\theta(x_t, t)$ and is trained using the standard noise prediction surrogate objective.

Optimization

- CIFAR10/CIFAR100 (32×32): batch size 128, learning rate 2×10^{-4}, 508K iterations.
- CelebA (64×64): batch size 128, learning rate 2×10^{-4}, 500K iterations.
- CelebAHQ (256×256): batch size 8, learning rate 2×10^{-5}, 500K iterations.

C.2.2 Improved Diffusion Models.

Improved diffusion models follow Nichol & Dhariwal (2021). They share the same U-Net backbones used for DDPM but differ by using a cosine noise schedule and a hybrid objective that learns both predicted noise and variance.

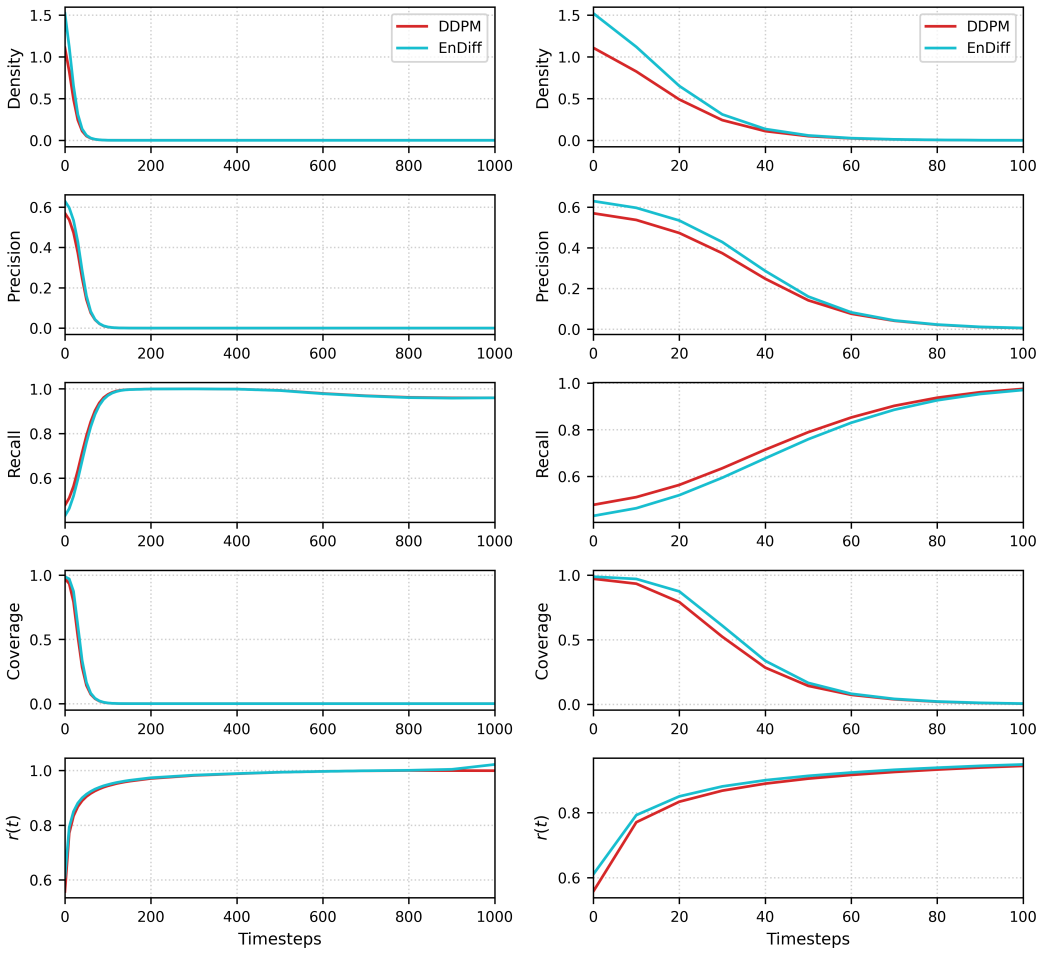


Figure 7: CIFAR10: PRDC metrics and predicted noise trace energy across diffusion timesteps. Left column shows the full timestep range; right column shows a zoomed view for early timesteps (t = 0-100). Rows report (top to bottom) Density, Precision, Recall, Coverage, and the predicted noise trace energy diagnostic $r(t)$, comparing DDPM and EnDiff.

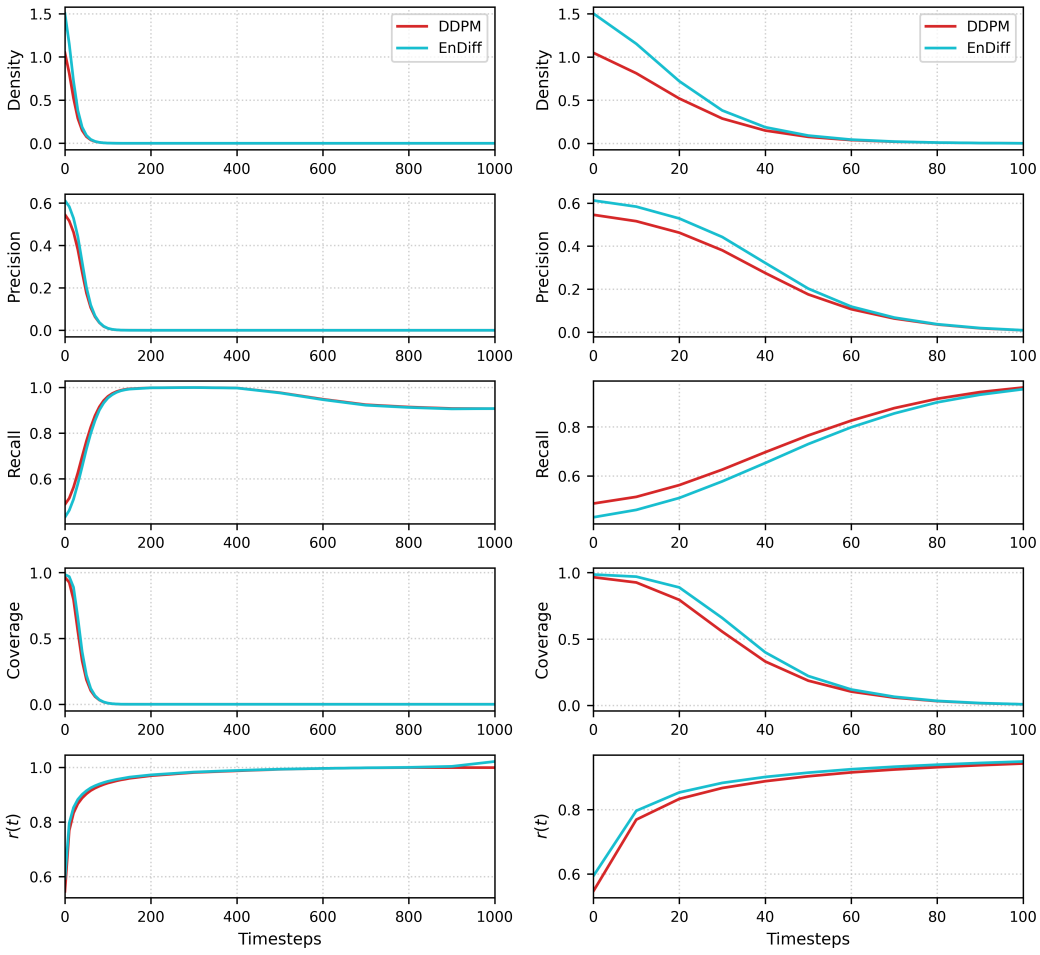


Figure 8: CIFAR100: PRDC metrics and predicted noise trace energy across diffusion timesteps. Left column shows the full timestep range; right column shows a zoomed view for early timesteps (t = 0-100). Rows report (top to bottom) Density, Precision, Recall, Coverage, and the predicted noise trace energy diagnostic $r(t)$, comparing DDPM and EnDiff.

Architecture The core U-Net is identical to that used in DDPM, with the number of residual blocks increased when specified. Attention layers were inserted at an intermediate spatial resolution vary by dataset:

- CIFAR10/CIFAR100 (32×32): $16 \times 16, 8 \times 8$ resolutions.
- CelebA (64×64): 16×16 resolution.
- CelebAHQ (256×256): 16×16 resolution.

Diffusion Setup All improved diffusion models use the cosine variance schedule and the hybrid noise prediction and learned variance objective. The number of diffusion steps remains 1000.

Optimization

- CIFAR10/CIFAR100 (32×32): batch size 128, learning rate 1×10^{-4}, 300K iterations, 3 residual blocks per resolution.
- CelebA (64×64): batch size 128, learning rate 1×10^{-4}, 500K iterations, 2 residual blocks per resolution.
- CelebAHQ (256×256): batch size 16, learning rate 2×10^{-5}, 500K iterations, 2 residual blocks per resolution.

C.2.3 Latent Diffusion Models.

Latent diffusion models follow Rombach et al. (2022) and operate in a learned latent space.

First-Stage Autoencoder LDM experiments use a trained AutoencoderKL. The latent prior is enforced via a KL divergence term. The autoencoder is trained separately and frozen for all diffusion experiments. Latent tensor and channel multipliers vary by dataset:

- CIFAR10/CIFAR100 (32×32): $8 \times 8 \times 3$ latent tensor, $[1, 2, 4]$ channel multipliers with 2 residual blocks per resolution.
- CelebA (64×64): $8 \times 8 \times 3$ latent tensor, $[1, 2, 4, 8]$ channel multipliers with 2 residual blocks per resolution.
- CelebAHQ (256×256): $64 \times 64 \times 3$ latent tensor, $[1, 2, 4]$ channel multipliers with 2 residual blocks per resolution.

Latent U-Net The diffusion U-Net in latent space uses a base channel width of 128 for CIFAR10/CIFAR100 and 192 for CelebA with channel multipliers $[1, 2, 4]$. For CelebAHQ, latent space uses a base channel width of 224 and channel multipliers $[1, 2, 3, 4]$. Attention is applied at latent resolutions 8, 4, and 2. Two residual blocks are used at each resolution. Dropout is not applied in any LDM configuration.

Diffusion and Optimization The diffusion process uses a linear schedule between $\beta_{\text{start}} = 0.0015$ and $\beta_{\text{end}} = 0.0195$. The loss is the standard simple L_2 noise prediction loss. Models were trained using the Adam optimizer. Training batch size, total iterations, and learning rate vary by dataset:

- CIFAR10/CIFAR100 (32×32): 128 batch size, 780K iterations, 1×10^{-4} learning rate
- CelebA (64×64): 64 batch size, 500K iterations, 2×10^{-7} learning rate.
- CelebAHQ (256×256): 128 batch size, 410K iterations, 2×10^{-7} learning rate.

C.3 Sampling Protocol

All reported generation metrics use DDPM ancestral sampling unless stated otherwise. We use 1000 inference steps for all datasets. For metric computation, we generate 50K samples for CIFAR10, CIFAR100, and CelebA, and 27K samples for CelebAHQ.

C.4 Evaluation Protocol

FID is computed using the TorchMetrics implementation with its default setting. For each dataset, we compute FID between generated samples and real images using matched counts: 50K real vs. 50K generated for CIFAR10/100 and CelebA, and 27K real vs. 27K generated for CelebAHQ.

PRDC is computed in the original image (pixel) space with neighborhood size $k = 5$. We report Precision, Recall, Density, and Coverage for all diffusion families (DDPM, Improved, LDM).

C.5 Class Conditional Generation

To test whether the objective level intervention extends beyond unconditional generation, we train a class conditional DDPM on CIFAR10 under the same architecture and training budget as the unconditional model, modifying only the training objective. Table 3 shows the same pattern as the unconditional results in the main text: Precision, Density, and Coverage improve while Recall decreases, and FID remains nearly unchanged.

Table 3: Class conditional generation on CIFAR10, **DDPM**. The baseline and its EnDiff$_{ddpm}$ variant use the same architecture and training budget, and differ only in the training objective.

Metric	Cond. DDPM	EnDiff$_{ddpm}$
FID ↓	**5.4411**	5.4455
Prec. ↑	0.7378	**0.7787**
Recall ↑	**0.3982**	0.3571
Dens. ↑	1.7571	**2.3195**
Cov. ↑	0.9927	**0.9977**

D Effect of Variance Schedules on 2D Synthetic Distributions and Sensitivity Analysis

D.1 Sensitivity Analysis of the Regularization Strength λ

As shown in Table 4, different values of λ generally improve Precision, Density, and Coverage.

Table 4: Sensitivity to the regularization weight λ on **CIFAR10**. Lower FID is better (↓); higher Precision, Recall, Density, and Coverage are better (↑).

λ	FID ↓	Precision ↑	Recall ↑	Density ↑	Coverage ↑
0.00	3.4430	0.6060	0.5320	1.0901	0.9704
0.01	3.4751	0.6103	0.5179	1.1832	0.9745
0.05	3.4567	0.6357	0.5288	1.2664	0.9793
0.10	3.6398	0.6221	0.4903	1.3569	0.9836
0.30	3.6445	0.6610	0.4849	1.4270	0.9850

D.2 Effect of Variance Schedules

As shown in tables 5 to 8, the qualitative effect of EnDiff remains broadly consistent across cosine, linear, quadratic, and sigmoid variance schedules for all four 2D synthetic distributions. Across all four 2D synthetic distributions, the qualitative effect of EnDiff remains broadly consistent under cosine, linear, quadratic, and sigmoid variance schedules. Relative to DDPM, EnDiff$_{ddpm}$ more consistently improves the fidelity oriented metrics, especially Precision and Density, while changes in Recall and Coverage are smaller and more schedule dependent. This indicates that the observed fidelity-diversity shift is not tied to a single schedule choice. At the same time, the magnitude of the improvement varies across distributions, showing that schedule design still influences the final trade off.

E Additional Qualitative Results

Figures 9 to 12 provide additional qualitative comparisons for the four image datasets.



Figure 9: CelebA-HQ (256×256) unconditional samples: DDPM (left) vs. EnDiff$_{ddpm}$ (right). EnDiff$_{ddpm}$ exhibits sharper local facial details and fewer structural artifacts, consistent with higher fidelity-oriented metrics.

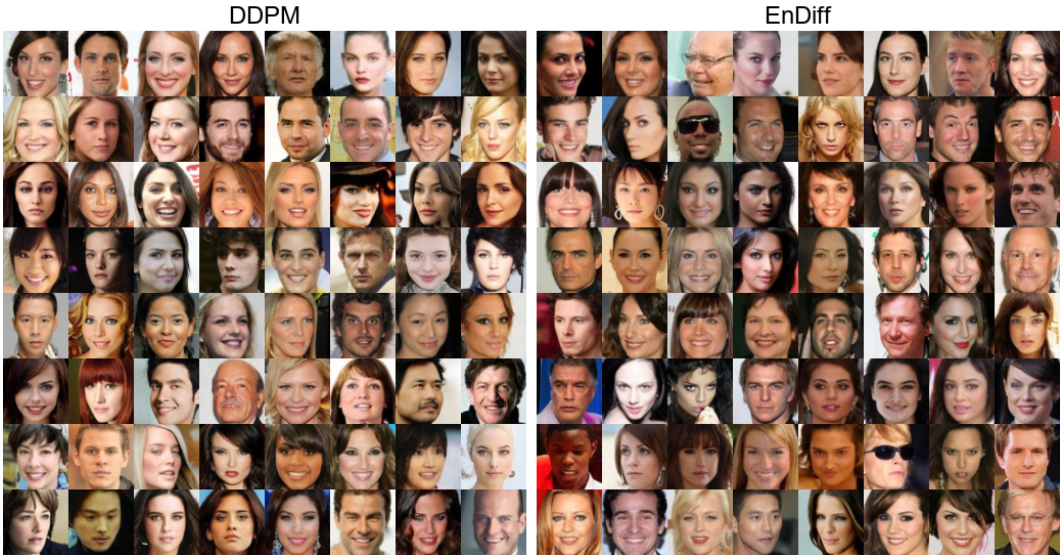


Figure 10: CelebA (64×64) unconditional samples: DDPM (left) vs. EnDiff$_{ddpm}$ (right). EnDiff$_{ddpm}$ yields more coherent facial structure and textures while maintaining broad attribute variety.



Figure 11: CIFAR10 (32×32) unconditional samples: DDPM (left) vs. EnDiff$_{ddpm}$ (right). EnDiff$_{ddpm}$ produces cleaner object boundaries and fewer local corruptions, aligning with improved Precision/Density.

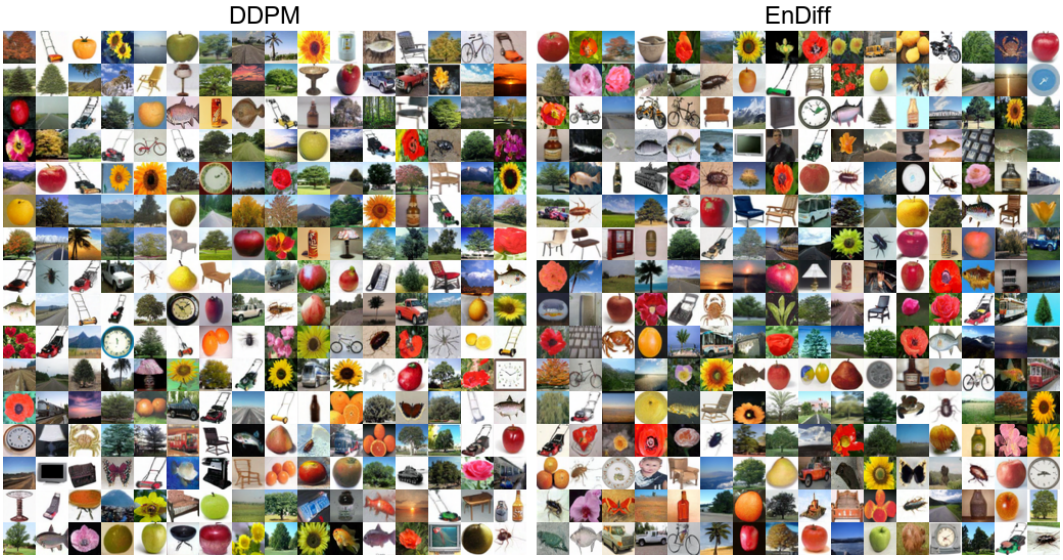


Figure 12: CIFAR100 (32×32) unconditional samples: DDPM (left) vs. EnDiff$_{ddpm}$ (right). EnDiff$_{ddpm}$ improves fine scale object structure and reduces spurious textures, consistent with the observed fidelity-diversity shift.

Table 5: Effect of variance schedule on PRDC for **Central Banana**. Higher is better (↑).

Schedule	Prec.↑	Rec.↑	Den.↑	Cov.↑
DDPM				
Cosine	0.971	**0.998**	0.960	**0.953**
Linear	0.970	0.998	0.964	**0.961**
Quadratic	0.976	**0.999**	0.974	**0.959**
Sigmoid	0.976	**0.999**	0.966	**0.949**
EnDiff$_{ddpm}$				
Cosine	**0.990**	0.997	**0.987**	0.952
Linear	**0.984**	**0.999**	**0.977**	0.954
Quadratic	**0.988**	0.997	**0.991**	0.954
Sigmoid	**0.990**	0.996	**0.993**	**0.949**

Table 6: Effect of variance schedule on PRDC for **Moon Circles**. Higher is better (↑).

Schedule	Prec.↑	Rec.↑	Den.↑	Cov.↑
DDPM				
Cosine	0.996	**0.998**	1.010	**0.966**
Linear	0.993	0.998	0.997	**0.964**
Quadratic	0.997	**0.997**	1.006	**0.965**
Sigmoid	0.997	**0.997**	1.009	0.961
EnDiff$_{ddpm}$				
Cosine	**0.998**	0.995	**1.012**	0.960
Linear	**0.995**	**0.999**	**1.007**	0.963
Quadratic	**0.998**	0.995	**1.011**	**0.965**
Sigmoid	**0.998**	**0.997**	**1.015**	**0.963**

Table 7: Effect of variance schedule on PRDC for **Scattered Moon**. Higher is better (↑).

Schedule	Prec.↑	Rec.↑	Den.↑	Cov.↑
DDPM				
Cosine	**0.999**	**0.997**	1.001	**0.967**
Linear	**0.999**	**0.997**	1.004	**0.966**
Quadratic	0.999	**0.998**	1.001	**0.967**
Sigmoid	0.999	**0.997**	0.998	**0.965**
EnDiff$_{ddpm}$				
Cosine	**0.999**	0.996	**1.010**	**0.967**
Linear	**0.999**	0.995	**1.006**	0.963
Quadratic	**1.000**	0.997	**1.006**	0.966
Sigmoid	**1.000**	0.996	**1.009**	0.963

Table 8: Effect of variance schedule on PRDC for **Swiss Roll**. Higher is better (↑).

Schedule	Prec.↑	Rec.↑	Den.↑	Cov.↑
DDPM				
Cosine	0.990	**0.998**	0.991	**0.961**
Linear	0.983	**0.999**	0.965	0.944
Quadratic	0.990	**0.997**	0.984	**0.959**
Sigmoid	0.991	**0.999**	0.990	**0.966**
EnDiff$_{ddpm}$				
Cosine	**0.994**	0.995	**1.009**	0.960
Linear	**0.986**	0.997	**0.989**	**0.953**
Quadratic	**0.993**	0.996	**1.012**	0.951
Sigmoid	**0.993**	0.991	**1.010**	0.954