Noise Scheduling as Information-Guided Allocation in Diffusion Training

Gabriel Raya[1*] **Bac Nguyen**[2] **Georgios Batzolis**[3] **Yuhta Takida**[2] **Dejan Stancevic**[4]
Naoki Murata[2] **Chieh-Hsin Lai**[2] **Yuki Mitsufuji**[2,5†] **Luca Ambrogioni**[4†]
[1]Tilburg University & JADS [2]Sony AI [3]University of Cambridge
[4]Radboud University [5]Sony Group Corporation.

Abstract

We introduce INFONOISE, an online adaptive noise schedule for diffusion training that reallocates optimization effort toward noise levels where denoising is most informative. Together with loss weighting, a noise schedule induces an effective allocation across denoising problems, often fixed before informative noise levels are known. INFONOISE makes this allocation data-adaptive by estimating a conditional-entropy-rate profile from denoising losses during training, without auxiliary models or offline search. Through I–MMSE, this profile identifies where noisy observations rapidly reduce uncertainty about the clean sample and guides adaptation of the training noise distribution. It changes only this distribution, keeping the objective, weighting, and parameterization fixed. On image benchmarks, where schedules have been extensively tuned, INFONOISE matches or slightly exceeds strong baselines and can reach the same quality with fewer updates. On representation, sequence, and modality shifts, including DNA and language generation, INFONOISE improves over fixed and adaptive baselines and reaches target quality with up to $3\times$ less training compute. These results establish the conditional-entropy-rate profile as the data-dependent target for noise schedule design and make online adaptation a practical alternative to manual schedule search.

1 Introduction

Diffusion models have become a dominant class of generative models, with strong performance across images, video, audio, language, and scientific data (Nichol and Dhariwal, 2021; Rombach et al., 2022; Ho et al., 2022; Kong et al., 2020; He et al., 2023; Kahouli et al., 2024; Lai et al., 2025). They generate data by learning to reverse a corruption process that gradually obscures structure with noise (Sohl-Dickstein et al., 2015; Song and Ermon, 2019; Ho et al., 2020; Song et al., 2020). Training this reverse process does not optimize a single denoising problem, but a weighted collection of denoising losses across noise levels. Each noise level induces a different posterior inference problem, with different remaining uncertainty and a different learning signal. The training noise schedule sets how often each problem is visited, while loss weighting sets how strongly its error enters the objective. Together they induce an *effective allocation* of training effort across the corruption path (Karras et al., 2022; Kingma and Gao, 2023). The central challenge is that this allocation is usually fixed before training reveals where along the path noisy observations are most informative.

In practice, diffusion training has often improved by changing this allocation indirectly. Noise schedules, loss weights, denoising objectives, and prediction parameterizations change which regions of the corruption path receive optimization effort (Nichol and Dhariwal, 2021; Karras et al., 2022; Hang et al., 2023; Kingma and Gao, 2023; Karras et al., 2024; Esser et al., 2024). Yet the informative region

*Work partially done during an internship at Sony AI, Tokyo, Japan.
†These authors contributed equally as senior authors.

Preprint.

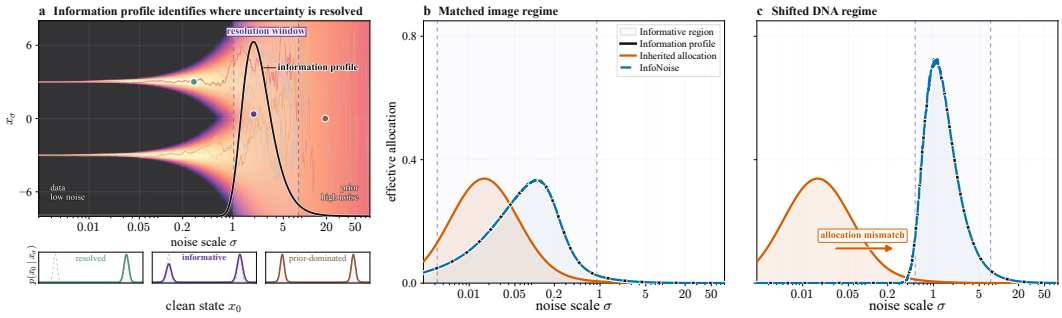


Figure 1: **The information profile defines the allocation target in diffusion training.** **(a)** In an analytical two-state Gaussian channel, uncertainty is not resolved uniformly along the path. The entropy-rate profile peaks at the decision window where noisy observations most rapidly reveal the clean state; the derivation is given in Section B.5. **(b)** In mature image regimes, inherited allocations already overlap the measured informative region, so INFONOISE preserves the allocation while recovering this region online. **(c)** Under shift, the information profile moves while inherited allocation remains fixed, producing allocation mismatch. INFONOISE estimates the shifted profile online and redirects allocation toward the region where uncertainty is resolved, without noise-schedule retuning.

can move with resolution, representation, data statistics, or modality, even within images (Hoogeboom et al., 2023); practice then relies on inherited schedules, empirical search, or objective-tied adaptive signals, while the allocation target remains implicit (Austin et al., 2021; He et al., 2023; Shi et al., 2024; Sahoo et al., 2024; Chen et al., 2026). When a schedule tuned in one regime is reused in another, it transfers the source allocation rather than measuring which noise levels are informative for the current training distribution. If the informative region shifts, inherited allocation remains tied to the source regime and spends updates where noisy observations resolve little uncertainty. We call this failure mode *allocation mismatch*. This raises the question of whether training allocation can adapt online to the data, rather than be inherited or recovered through empirical search.

The allocation target can be made precise by asking where uncertainty is resolved along the corruption path. Conditional entropy measures the uncertainty about the clean sample that remains after a noisy observation (Shannon, 1948); its pathwise rate identifies where observations reduce that uncertainty most rapidly. For affine-Gaussian corruption paths, I–MMSE links this rate to Bayes-optimal denoising error (Guo et al., 2005; Palomar and Verdu, 2006). The resulting conditional-entropy-rate profile gives a data-dependent target for training allocation. Schedule tuning can be understood as an indirect search for this profile. Fixed schedules are effective when they place effort near it, and inefficient when the profile shifts while allocation remains inherited. Thus, the governing object for schedule design is not the schedule family itself, but the information profile over noise levels.

We introduce INFONOISE, an online noise-schedule adaptation method that estimates this information profile from denoising losses during training and updates only the training noise distribution. The objective, loss weighting, and prediction parameterization remain fixed, so the comparison isolates training allocation from changes to the learning problem. Unlike entropic-time methods that reparameterize inference-time discretization of a pretrained model (Stancevic et al., 2026), INFONOISE adapts the training allocation itself. Figure 1 illustrates the allocation view. Along the corruption path, uncertainty about the clean sample is resolved non-uniformly, so efficient training should concentrate effort where denoising carries the most information. Panel (a) shows a controlled two-state Gaussian channel in which the conditional-entropy-rate profile localizes the resolution window. Panel (b) shows the matched case, where tuned image allocations overlap the informative region and INFONOISE recovers this region online. Panel (c) shows the shifted case, where the informative region moves while inherited allocation remains fixed, producing allocation mismatch. INFONOISE uses the recovered information profile to adapt training to the denoising problem itself, rather than to an inherited schedule, moving updates toward the regions where uncertainty is resolved.

Contributions. **(i)** We formalize diffusion noise scheduling as an effective allocation problem along the corruption path. **(ii)** We identify the conditional-entropy-rate profile as the data-dependent target for this allocation, explaining when fixed schedules transfer and when they misallocate compute. **(iii)** We introduce INFONOISE, an online adaptive noise schedule that estimates this profile from denoising losses and updates only the training noise distribution, replacing repeated schedule search with online information-guided allocation.

2 Effective allocation and the information profile

Let p_{data} denote the distribution of clean samples on $\mathbb{R}^d$, and let $\mathbf{x}_0 \sim p_{\text{data}}$. Diffusion models define a corruption path that turns clean samples into noisy observations and train a reverse model through the denoising problems induced along it. The path is often parameterized by $t \in [0, 1]$ and a corruption schedule, such as a noise scale $\sigma(t)$, a monotone transform of log-SNR, or signal and noise coefficients $a(t)$ and $b(t)$. For allocation, the relevant object is the ordered family of denoising problems, not the coordinate used to label it. We write u for the coordinate in which allocation is analyzed, chosen in the data-to-noise direction. For the affine-Gaussian paths considered here,

$$\mathbf{x}_u = a(u)\mathbf{x}_0 + b(u)\boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{1}$$

where $a(u)$ and $b(u)$ set the signal and noise strengths. For all path values used below, $a(u), b(u) > 0$, and the signal-to-noise ratio $\gamma(u) = a(u)^2/b(u)^2$ is non-increasing in u. Each value of u defines the posterior denoising problem of estimating $\mathbf{x}_0$ from $\mathbf{x}_u$. In VE paths, $u = \sigma$, $a(u) \equiv 1$, and $b(u) = u$, giving $\mathbf{x}_\sigma = \mathbf{x}_0 + \sigma\boldsymbol{\varepsilon}$. Standard VE, VP, and sub-VP corruption kernels fit this form, as do the Gaussian conditional interpolants used in our flow-matching baselines (Song et al., 2020; Karras et al., 2022; Kingma and Gao, 2023; Lipman et al., 2023).

2.1 Effective allocation

Let $\pi(u)$ denote the training density over path locations in the coordinate used to compare allocation. When training samples a different monotone coordinate s, with $s \sim \pi_s$ and $u = g(s)$, the induced density over u is

$$\pi(u) = \pi_s(s(u)) \left| \frac{\mathrm{d}s}{\mathrm{d}u} \right|. \tag{2}$$

A monotone change of coordinate on a fixed path leaves the corrupted observations and posterior denoising problems unchanged, but changes the training density induced over them. After rewriting the loss in an x-prediction view, define the expected denoising loss at path location u as

$$R_x(u; \theta) = \mathbb{E}_{\mathbf{x}_0 \sim p_{\text{data}},\, \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[\|\mathbf{x}_0 - \hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{x}_u; u)\|_2^2 \right]. \tag{3}$$

Following the design-space view of Karras et al. (2022) and the weighted-objective view of Kingma and Gao (2023), common diffusion losses can be written as weighted denoising objectives after conversion to a common prediction space. The training density $\pi(u)$ and the induced per-noise weight $w(u) \geq 0$ therefore jointly determine the effective allocation. The conversions used in our baselines are given in Appendix A. The weighted objective is

$$\mathcal{L}(\theta) = \int \underbrace{\pi(u)\, w(u)}_{\phi(u)}\, R_x(u; \theta)\, \mathrm{d}u. \tag{4}$$

Here $\pi(u)$ determines how often u is sampled, while $w(u)$ sets the objective weight of the expected denoising loss at that location. We call $\phi(u) \coloneqq \pi(u)w(u)$ the unnormalized effective allocation. After normalization, ϕ gives the allocation profile over denoising problems.

2.2 Information profile

The effective allocation ϕ specifies where the objective places training weight along the corruption path. The target for this allocation is determined by the posterior family $p(\mathbf{x}_0 \mid \mathbf{x}_u)$. Its residual uncertainty is

$$\mathrm{H}[\mathbf{x}_0 \mid \mathbf{x}_u] \triangleq -\mathbb{E}_{p(\mathbf{x}_0, \mathbf{x}_u)} \left[\log p(\mathbf{x}_0 \mid \mathbf{x}_u) \right]. \tag{5}$$

Since $\mathrm{H}[\mathbf{x}_0]$ is fixed along the path,

$$I(\mathbf{x}_0; \mathbf{x}_u) = \mathrm{H}[\mathbf{x}_0] - \mathrm{H}[\mathbf{x}_0 \mid \mathbf{x}_u], \tag{6}$$

and therefore

$$\frac{\mathrm{d}}{\mathrm{d}u}\mathrm{H}[\mathbf{x}_0 \mid \mathbf{x}_u] = -\frac{\mathrm{d}}{\mathrm{d}u} I(\mathbf{x}_0; \mathbf{x}_u). \tag{7}$$

The target is a pathwise rate, not an uncertainty level. It marks where the forward path loses mutual information and where the reverse model must reconstruct it.

At path location u, the MSE-optimal x-prediction denoiser is the posterior mean $\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_u]$. The corresponding unweighted Bayes risk is

$$\mathrm{mmse}(u) \triangleq \mathbb{E}\Big[\|\mathbf{x}_0 - \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_u]\|_2^2\Big] = \mathbb{E}[\mathrm{tr}\,\mathrm{Cov}(\mathbf{x}_0 \mid \mathbf{x}_u)]\,. \tag{8}$$

MMSE measures residual posterior uncertainty at a fixed path location. The allocation target is not this uncertainty level, but its rate of change along the path. Regions with small entropy rate are weak allocation targets, either because the posterior is already resolved or because neighboring path locations remain similarly unresolved.

For affine-Gaussian paths, standardizing the channel gives

$$\frac{\mathbf{x}_u}{b(u)} = \sqrt{\gamma(u)}\,\mathbf{x}_0 + \boldsymbol{\varepsilon}, \qquad \gamma(u) = \frac{a(u)^2}{b(u)^2}. \tag{9}$$

The Gaussian-channel I–MMSE identity (Guo et al., 2005; Palomar and Verdu, 2006) then yields

$$\frac{\mathrm{d}}{\mathrm{d}u}\mathrm{H}[\mathbf{x}_0 \mid \mathbf{x}_u] = -\frac{1}{2}\,\gamma'(u)\;\mathrm{mmse}(u). \tag{10}$$

With u ordered from data to noise, $\gamma'(u) \le 0$. We define the information profile $\rho^\star(u)$ as the nonnegative entropy rate

$$\rho^\star(u) \;\propto\; \frac{\mathrm{d}}{\mathrm{d}u}\mathrm{H}[\mathbf{x}_0 \mid \mathbf{x}_u] = -\frac{1}{2}\gamma'(u)\,\mathrm{mmse}(u). \tag{11}$$

The factor $|\gamma'(u)|$ converts Bayes denoising difficulty into a rate in the chosen coordinate. The profile is therefore neither raw training loss nor MMSE alone. It identifies where posterior uncertainty changes fastest along the corruption path, and gives the principled target for the effective allocation ϕ. This profile is the Bayes-limit counterpart of the SNR-rate denoising integrand in continuous-time VDM (Kingma et al., 2021). The full rewriting is given in Section A.3.

Figure 2 shows this distinction in a symmetric two-point VE channel. MMSE tracks residual posterior uncertainty, while the entropy-rate profile localizes where that uncertainty changes. Details are given in Section B.5.

Noise schedule design can therefore be stated as alignment between the effective allocation and the information profile. For fixed w, changing the sampling density π changes the induced allocation $\phi(u) = \pi(u)w(u)$. Both ϕ and $\rho^\star$ are coordinate-dependent profiles. After normalization, they transform as densities under monotone reparameterizations, so alignment is meaningful only in a common coordinate.

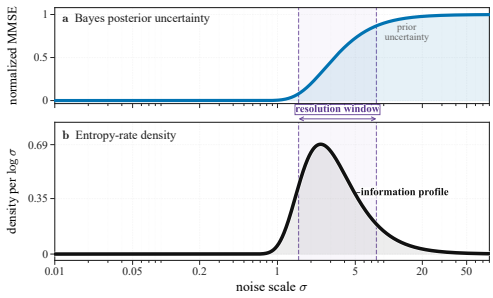


Figure 2: **Entropy rate localizes uncertainty resolution.** MMSE measures residual posterior uncertainty; entropy rate identifies where that uncertainty changes along the path.

VE specialization. For VE paths, choosing $u = \sigma$ gives $\mathbf{x}_\sigma = \mathbf{x}_0 + \sigma\boldsymbol{\varepsilon}$ and $\gamma(\sigma) = \sigma^{-2}$. Hence,

$$\frac{\mathrm{d}}{\mathrm{d}\sigma}\mathrm{H}[\mathbf{x}_0 \mid \mathbf{x}_\sigma] = \frac{\mathrm{mmse}(\sigma)}{\sigma^3} \;\ge\; 0. \tag{12}$$

The VE form makes the coordinate dependence explicit. MMSE measures how much posterior uncertainty remains at noise level σ, while $\mathrm{mmse}(\sigma)/\sigma^3$ measures where that uncertainty is being resolved along the VE path. Offline profiles provide diagnostic comparisons of allocation. Expressions for VE, VP, sub-VP, and affine-Gaussian flow-matching paths are summarized in Table 3 and derived in Appendix A.

3 INFONOISE as online information-guided allocation

The previous section identifies the information profile as the target for effective allocation. This target depends on the Bayes MMSE and is not available before training. INFONOISE estimates the profile online from unweighted denoising losses and changes only the training noise distribution. Since the induced allocation is $\phi(u) = \pi(u)w(u)$, the training noise distribution is updated so that $\pi(u)w(u)$ follows the estimated profile. The objective, loss weighting, prediction parameterization, architecture, and optimizer remain fixed.

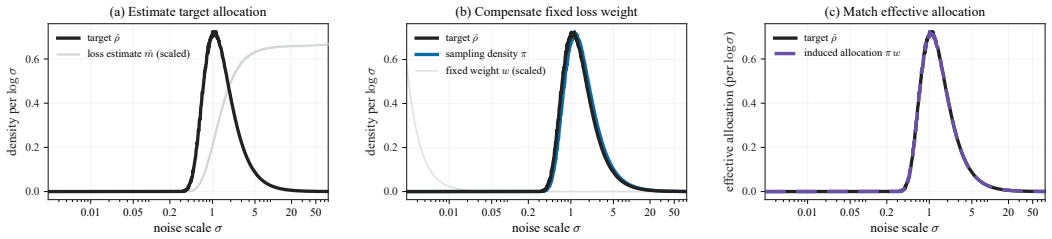


Figure 4: **INFONOISE converts an online information estimate into a training noise distribution.** **(A)** Unweighted losses track x-space denoising error, which the path factor converts into an online information-profile estimate. **(B)** With fixed loss weights, INFONOISE samples from $\pi \propto \hat{\rho}/w$, not directly from $\hat{\rho}$. **(C)** The induced allocation πw matches the profile while the objective and weighting remain unchanged. Noise levels are drawn from π by inverse-CDF sampling, illustrated in Figure 3.

3.1 Online estimation of the information profile

From Section 2.2, the target profile satisfies

$$\rho^\star(u) \propto \frac{1}{2}|\gamma'(u)|\,\mathrm{mmse}(u).$$

The Bayes MMSE is unavailable during training. INFONOISE maintains a smoothed binwise estimate $\hat{m}(u)$ from the unweighted x-space denoising losses already computed in each batch. The raw profile estimate is

$$\hat{q}(u) \;\propto\; \frac{1}{2}\,|\gamma'(u)|\,\hat{m}(u), \qquad \hat{q}(\sigma) \propto \frac{\hat{m}(\sigma)}{\sigma^3} \quad \text{for VE.} \tag{13}$$

The unweighted loss provides an online statistic for the denoising-error term, while the path factor converts this statistic into an entropy-rate estimate.

Before normalization, the estimate is gated near the low-noise endpoint in the training coordinate,

$$r(u) = \hat{q}(u)\,g_{c,n}(u), \qquad g_{c,n}(u) = \frac{u^n}{u^n + c^n}, \qquad \hat{\rho}_u(u) = \frac{r(u)}{\int r(s)\,\mathrm{d}s}. \tag{14}$$

The gate prevents the path factor from amplifying unstable endpoint estimates during early training, while leaving the interior profile unchanged for $u \gg c$. Smoothing and periodic refreshes further stabilize the estimate.

3.2 Converting the target profile into a sampler

The loss weight $w(u)$ is fixed by the training objective. Matching the estimated profile requires the induced allocation to satisfy $\pi(u)w(u) \propto \hat{\rho}_u(u)$, which gives

$$\pi(u) = \frac{\hat{\rho}_u(u)/w(u)}{\int \hat{\rho}_u(s)/w(s)\,\mathrm{d}s}, \qquad w(u) > 0. \tag{15}$$

The resulting allocation satisfies $\phi(u) = \pi(u)w(u) \propto \hat{\rho}_u(u)$. INFONOISE does not reweight the loss. It changes how often the fixed weighted objective visits each denoising problem. Figure 4 summarizes the conversion from online loss statistics to target profile, sampling density, and induced allocation.

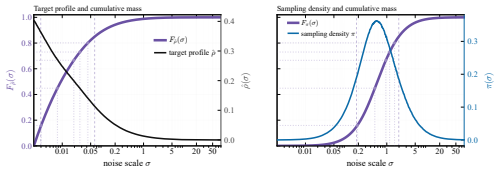


Figure 3: **Inverse-CDF sampling from the adapted density.** Shown on CIFAR-10 for clarity, the adapted density π defines a cumulative mass F_π. Uniform samples are mapped through F_π^{-1} to draw training noise levels. The same construction applies unchanged in other domains.

Sampling from π uses

$$F_\pi(u) = \int_{u_{\min}}^{u} \pi(s)\,\mathrm{d}s, \qquad \xi \sim \mathrm{Uniform}(0,1), \qquad u = F_\pi^{-1}(\xi). \tag{16}$$

In practice, F_π is tabulated on a fixed grid and inverted by interpolation, as illustrated in Figure 3.

3.3 Online adaptation loop

Algorithm 1 summarizes the online update. The estimator is maintained on a fixed grid in the training coordinate u. Each batch contributes the unweighted denoising loss ℓ to the profile estimate, while the model update still uses the original weighted loss $w(u)\ell$. The estimator changes the sampler, not the objective.

Warm-up uses a fixed prior π_0 to collect initial loss statistics. After the first refresh, the sampler is rebuilt from $\hat{\rho}_u(u)/w(u)$, so the induced allocation follows the learned profile rather than the warm-up prior. The estimator, gate, smoothing rule, and refresh cadence remain fixed. Only the sampling distribution adapts online.

Algorithm 1: INFONOISE online adaptation

Initialize $\hat{m}(u) \equiv 1$, choose warm-up prior $\pi_0(u)$, set $\pi \leftarrow \pi_0$;
for *training steps* **do**
  Sample $\mathbf{x}_0 \sim p_{\text{data}}$, $u \sim \pi(u)$, $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and set $\mathbf{x}_u = a(u)\mathbf{x}_0 + b(u)\boldsymbol{\varepsilon}$;
  Compute $\ell = \|\mathbf{x}_0 - \hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{x}_u; u)\|_2^2$, update model with $w(u)\ell$, and update $\hat{m}$ with (u, ℓ);
  if *refresh step* **then**
    Smooth $\hat{m}$, set $\hat{q}(u) \leftarrow \frac{1}{2}|\gamma'(u)|\hat{m}(u)$, gate and normalize to obtain $\hat{\rho}_u$, then update $\pi(u) \propto \hat{\rho}_u(u)/w(u)$;

4 Experiments

We now present empirical evidence that the information profile can serve as an automatic target for noise-schedule design. Standard image diffusion provides the already-optimized regime. Schedules such as EDM and cosine have been refined through extensive practice, so their allocations provide strong fixed references. In this regime, adaptation should preserve matched-budget quality rather than disturb an already useful allocation. The harder case is transfer. When the denoising problem changes through representation, sequence structure, or modality, inherited schedules need not place training effort near the informative region. We therefore evaluate mature image benchmarks as optimized regimes, and binarized images, digitized images, DNA regulatory sequences following the DNA-Diffusion setup (DaSilva et al., 2026), and text as shifted regimes. The expected pattern is asymmetric. Information-guided adaptation should preserve matched-budget quality when inherited allocations are aligned and reduce training compute when the informative region moves.

Experimental design. All comparisons isolate training allocation. Schedules are expressed in the common x-prediction view, where sampling density and loss weight induce an effective allocation over denoising problems (Kingma and Gao, 2023). Within each domain, only the training noise distribution varies. The model, objective, loss weighting, optimizer, EMA, augmentations, inference sampler, and evaluation budget are fixed. Fixed baselines include image-tuned EDM sampling (Karras et al., 2022), variance-preserving cosine sampling (Nichol and Dhariwal, 2021), flow-matching OT sampling (Lipman et al., 2023), and log-uniform sampling in σ. KG-adaptive is the adaptive control, following weighted-loss magnitude rather than the entropy-rate profile (Kingma and Gao, 2023). Offline profiles are used only for alignment analysis and are never provided to INFONOISE during training. Dataset-specific metrics are reported in Table 1. Schedule conversions are given in Appendix A; architectures, metrics, training details, and settings are given in Section C.

4.1 Information-guided allocation preserves tuned regimes and accelerates transfer

Table 1 and Figure 5 show that information-guided allocation can reproduce the benefit of tuned image schedules and adapt online during training when those schedules no longer match the denoising problem, without additional schedule retuning. On MNIST, FashionMNIST, CIFAR-10, and FFHQ, fixed schedules such as EDM and cosine are strong references because their allocations have been refined through empirical practice. In these optimized regimes, INFONOISE preserves matched-budget quality against the strongest baselines and often reaches the same target with fewer updates. The more demanding case is transfer. When representation, sequence structure, or modality changes, inherited image schedules need not place training effort near the informative region. On binarized images, digitized CIFAR-10, DNA, and OpenWebText-2, INFONOISE matches or improves final quality while reaching the strongest non-INFONOISE baseline target with up to $3\times$ less training compute. The information profile therefore provides an online target for schedule design, replacing manual search over fixed schedule families across these settings. The next subsection tests this mechanism by comparing effective allocations with measured information profiles.

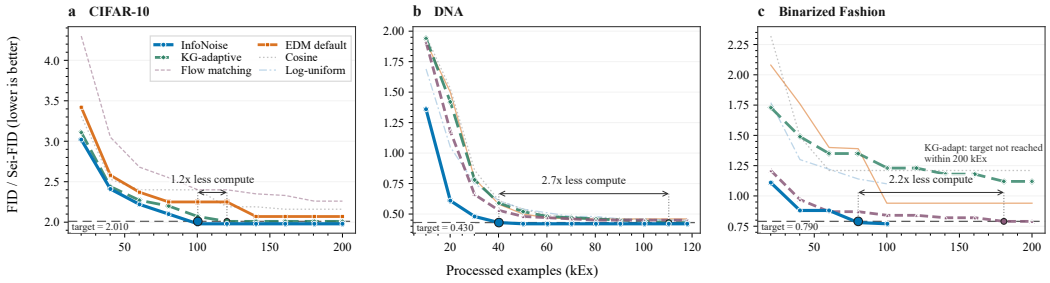


Figure 5: **INFONOISE reaches the best baseline quality with fewer training updates.** For each dataset, the target is the final matched-budget quality of the strongest non-INFONOISE baseline, including fixed schedules and KG-adaptive. Speedup reports how much less training compute INFONOISE needs to reach that target. A value of 1.00 means no measured reduction within the training budget; values above 1.00 indicate that the same quality is reached with fewer updates.

			Fixed schedules				Adaptive schedules		Efficiency
Dataset	Metric	NFE	Log-unif.	Cosine	FM-OT	EDM	KG-adapt.	INFONOISE	× vs fixed
MNIST	FID ↓	35	1.02	0.46	0.44	0.44	**0.39**	0.41	**1.20**
FashionMNIST	FID ↓	35	2.75	1.76	1.93	1.78	1.82	**1.71**	**1.45**
CIFAR-10 uncond.	FID ↓	35	5.93	2.16	2.31	2.04	2.00	**1.98**	**1.40**
CIFAR-10 cond.	FID ↓	35	4.45	2.11	2.13	1.85	**1.84**	**1.84**	**1.50**
FFHQ 64×64	FID ↓	79	4.68	2.69	2.72	**2.53**	2.54	**2.53**	**1.00**
bMNIST	FID ↓	35	0.63	0.52	0.38	0.53	0.38	**0.33**	**2.20**
bFashionMNIST	FID ↓	35	1.10	1.20	0.80	0.94	1.22	**0.69**	**3.00**
Digitized CIFAR-10	FID ↓	255	17.30	—	—	9.90	—	**3.10**	**3.00**
DNA	Sei-FID ↓	35	0.61	**0.42**	0.44	0.45	0.43	**0.42**	**2.00**
OpenWebText-2	MAUVE ↑	127	0.68	—	—	0.83	—	**0.87**	**2.06**

Table 1: **Information-guided allocation automates schedule adaptation across regimes.** Each row compares schedules trained to the same budget and evaluated at the listed NFE. Quality is the final metric at that budget. Efficiency reports the compute INFONOISE needs to reach the final metric of the best competing baseline; values above one mean the same target is reached with fewer updates. Shading denotes INFONOISE; bold marks best final quality and speedups above one.

4.2 Profile alignment explains schedule transfer

Figure 6 compares the measured information profile with the effective allocation induced by inherited EDM sampling and by INFONOISE. The profiles are computed after training from reference denoisers, used only for analysis, and never provided to INFONOISE during training. In CIFAR-10 and FFHQ, the inherited image allocation already overlaps the informative region. INFONOISE recovers a similar allocation online, showing that the online information estimate can recover the region that image schedule design has found empirically. Under representation, sequence, and modality shift, this overlap no longer holds. In binarized images, DNA, and text, the fixed schedule continues to spend updates according to the image regime, while INFONOISE moves the induced allocation toward the shifted profile without schedule retuning. The largest speedups in Table 1 occur in these displaced-profile regimes. Matched image regimes therefore test automatic recovery of a tuned allocation; shifted regimes test adaptation when that allocation no longer follows the denoising difficulty.

4.3 Schedule search is allocation search

When a schedule is transferred to a new domain, the usual remedy is to retune its parameters by training several candidates and selecting the best validation run. In the allocation view, this procedure is not an abstract hyperparameter search. It is a search over fixed effective allocations. The EDM log-normal training distribution provides a controlled instance of this search. Holding the objective, loss weighting, prediction parameterization, model, optimizer, sampler, and evaluation fixed, changing P_{mean} in $\log \sigma \sim \mathcal{N}(P_{\mathrm{mean}}, P_{\mathrm{std}}^2)$, with P_{std} fixed, shifts only the sampling density π, and therefore

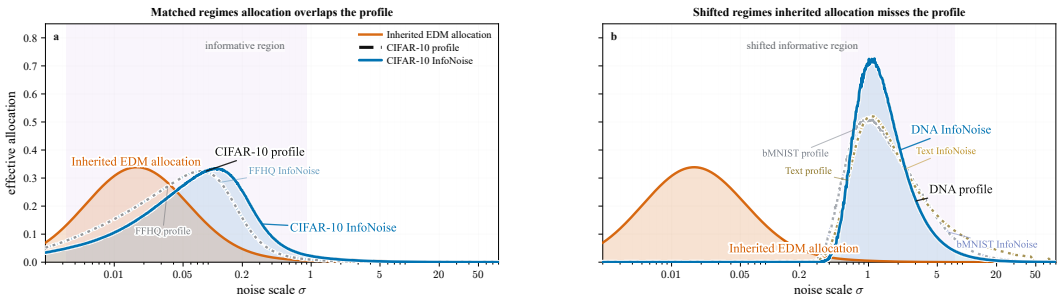


Figure 6: **Profile alignment explains schedule transfer.** Post hoc information profiles are compared with the effective allocation induced by inherited EDM sampling and by INFONOISE. **(A)** In matched image regimes, inherited allocation overlaps the informative region, and INFONOISE converges to a similar allocation online. **(B)** Under shift, the information profile moves away from inherited image allocation. INFONOISE moves the induced allocation toward the shifted profile without schedule retuning. Profiles are used only for analysis and never provided to INFONOISE during training.

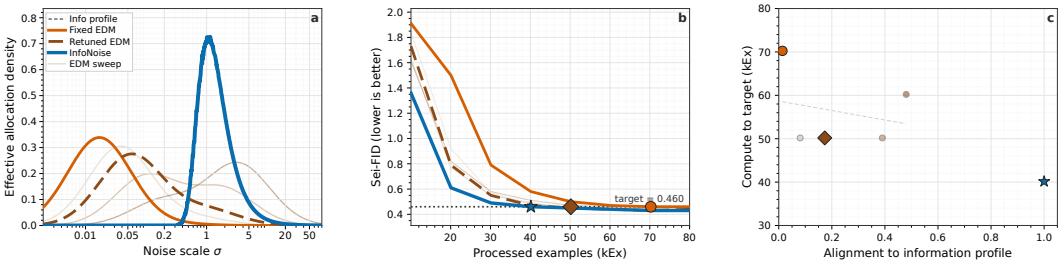


Figure 7: **Schedule search is allocation search.** **(a)** Sweeping P_{mean} in the EDM log-normal schedule shifts effective allocation along the DNA corruption path while other training choices remain fixed. INFONOISE reaches the informative region online in one run. **(b)** Allocations closer to the information profile reach the strongest non-INFONOISE baseline target with fewer updates. **(c)** Training compute decreases as effective allocation becomes better aligned with the DNA information profile, showing that misalignment with the profile is what makes schedule transfer expensive.

the effective allocation $\phi = \pi w$. DNA is a shifted regime where the inherited image allocation lies away from the measured information profile. Figure 7 shows that moving P_{mean} moves allocation along the DNA corruption path, and that performance improves as this allocation approaches the informative region. Settings that remain near the inherited image allocation converge more slowly and reach worse Sei-FID; settings closer to the profile reach the target with less compute. INFONOISE reaches the same region in a single run by adapting allocation toward the information profile, replacing repeated schedule trials with information-guided noise allocation.

4.4 Online adaptation overrides warm-up and stabilizes

Adaptive sampling starts from a prior only because no profile estimate exists at initialization. On DNA, Figure 9 shows that this prior is quickly superseded. The online estimate moves from the warm-up allocation toward the post hoc diagnostic information profile within the first rebuilds, while the sampler rebuilds the induced effective allocation around this estimate. The diagnostic profile is computed only after training from a reference denoiser and is never provided to INFONOISE. Consecutive sampler-CDF changes fall below 10^{-2} after the first rebuilds, indicating that the sampler stabilizes rather than following transient loss fluctuations. Table 2 gives the corresponding endpoint check. Changing only the warm-up prior across EDM, log-uniform, and log-σ constant-entropy starts changes final Sei-FID by 0.007. The DNA result is therefore driven by the online profile update after warm-up, not by the initial sampling prior.

Warm-up prior	Sei-FID ↓
EDM log-normal	0.460
Log-uniform in σ	0.463
Log-σ const.-entropy	**0.456**
Spread	0.007

Table 2: **Warm-up does not set final performance.** DNA Sei-FID changes by only 0.007 across warm-up priors.

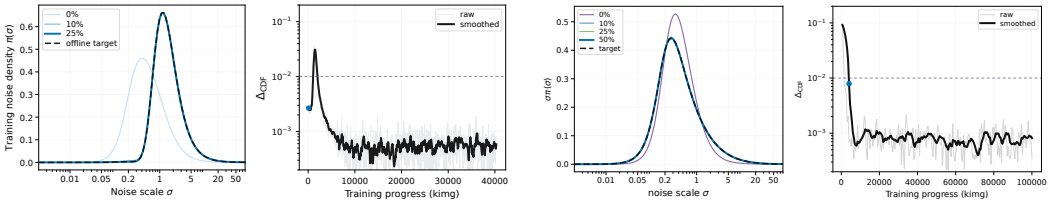


Figure 9: **INFONOISE rapidly recovers the information profile and stabilizes.** On DNA, the online estimate moves from the warm-up allocation toward the post hoc diagnostic profile within the first rebuilds. The diagnostic profile is computed after training and never provided to INFONOISE. Sampler-CDF changes fall below 10^{-2}, showing that the sampler stabilizes quickly rather than tracking transient loss fluctuations.

4.5 Training-allocation gains persist under guidance

Classifier-free guidance changes how the trained conditional and unconditional predictions are combined at sampling time. We keep this inference procedure fixed across schedules by using the same guidance scale, sampler, and generation budget. The CFG training recipe is also fixed, including the conditional-dropout setup. Thus the only difference between runs is the training noise distribution. Figure 8 shows that INFONOISE reaches higher MAUVE earlier under the same guided sampler. The allocation learned during training therefore remains beneficial under guided inference, rather than depending on guidance-scale or sampler tuning.

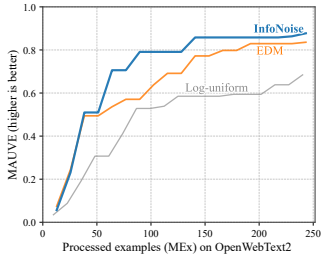


Figure 8: **Training-allocation gains persist under fixed CFG.** With guidance scale and sampler fixed, INFONOISE reaches higher MAUVE earlier.

5 Related work

Noise schedules, weighting, and training allocation. Much of diffusion training design can be read as deciding where denoising errors should matter along the corruption path. Noise schedules determine which path locations are visited, while loss weights and prediction parameterizations determine how their errors enter the objective. Early diffusion and score-based models fixed this choice through variance schedules, log-spaced noise scales, and noise-dependent objectives (Sohl-Dickstein et al., 2015; Song and Ermon, 2019; Ho et al., 2020; Song et al., 2020). Later work refined the allocation through cosine and learned schedules, log-SNR parameterizations, SNR-aware weighting, importance sampling, and loss-aware sampling (Nichol and Dhariwal, 2021; Kingma et al., 2021; Karras et al., 2022; Kingma and Gao, 2023; Hang et al., 2023; Hoogeboom et al., 2023; Chen, 2023). Spectral approaches make schedule ranges instance-dependent for pixel diffusion by using image frequency content to remove redundant noise levels and improve low-step sampling (Esteves and Makadia, 2026). EDM separates path, preconditioning, weighting, sampler, parameterization, and training noise distribution, but the training distribution remains chosen beforehand or retuned across runs (Karras et al., 2022). Flow matching and rectified flows expose the same dependence on path-time sampling (Lipman et al., 2023; Esser et al., 2024). Discrete and language diffusion further expose schedule-transfer limits through absorbing masks, mutual-information schedules, state-dependent masking, contextual schedulers, and schedule-conditioned transitions (Austin et al., 2021; Dieleman et al., 2022; He et al., 2023; Shi et al., 2024; Sahoo et al., 2024; Chen et al., 2026). INFONOISE keeps the objective, parameterization, and loss weight fixed, and adapts only the training noise distribution so that the induced allocation follows an online estimate of the information profile.

Information transfer as a schedule-design target. The allocation induced by a training objective needs a target. For Gaussian channels, I–MMSE links information-change rate to Bayes denoising error (Guo et al., 2005; Palomar and Verdu, 2006). This identity has been used for diffusion objectives, likelihoods, regularization, and path geometry (Kong et al., 2023; Franzese et al., 2024; Wang et al., 2023; Premkumar, 2026; Wang et al., 2025; Stančević and Ambrogioni, 2026). Entropy-based schedulers use information signals for time reparameterization or inference-step allocation (Stancevic et al., 2026), and LangFlow uses information gain for language-diffusion schedules (Chen et al., 2026). INFONOISE brings the conditional-entropy-rate profile to training, estimates it online from denoising losses, and uses it as the target for the effective allocation induced by sampling and weighting.

Localized denoising difficulty and critical regions. Denoising difficulty is not uniform along the generative path. Target-field variance can concentrate in restricted regions, motivating modified targets, reference information, or changes to the learned field (Xu et al., 2023; Yang et al., 2026). Work on symmetry breaking and critical dynamics shows a complementary form of localization. Diffusion trajectories do not recover structure gradually and uniformly. They pass through windows where symmetries break, commitments form, and sample structure emerges over a restricted range of noise levels (Raya and Ambrogioni, 2023; Biroli et al., 2024; Li and Chen, 2024; Sclocchi et al., 2025). These results make noise level choice part of the mechanism of pattern formation, not only a numerical discretization choice. In our allocation view, this localization appears as a profile of uncertainty resolution along the corruption path. INFONOISE uses this profile as a training-side allocation target while keeping the model, objective, parameterization, and sampler fixed.

6 Discussion and conclusions

Diffusion training allocates optimization effort across denoising problems. The allocation view puts noise schedules and loss weights into one object, the effective allocation along the corruption path. This reframes noise-schedule design as deciding where training effort should be placed, rather than selecting a fixed schedule family inherited from another regime. The information profile provides the target. For affine-Gaussian corruption paths, this profile is the conditional-entropy rate, linked by I–MMSE to Bayes denoising difficulty. INFONOISE estimates this profile online and adapts only the training noise distribution, while leaving the objective, loss weighting, parameterization, architecture, optimizer, and sampler fixed. The empirical results follow this structure. When inherited allocations match the profile, as in optimized image regimes, INFONOISE preserves matched-budget quality. When the profile shifts under representation, sequence, or modality change, information-guided adaptation redirects allocation and reaches strong baseline targets with less compute. The alignment, schedule-search, online-recovery, and warm-up analyses support a single interpretation. The gains come from adapting effective allocation toward the information profile, not from loss-magnitude resampling, favorable initialization, or post hoc access to the profile. Noise-schedule design should therefore move from retuning inherited schedule families to estimating the task-specific information structure of the denoising problem. Fixed schedules remain useful when their allocation matches this structure. When the profile moves, schedule search becomes an indirect way of rediscovering where uncertainty is resolved. INFONOISE makes that structure available during training and turns schedule design into online information-guided allocation matched to the task-specific denoising problem.

Limitations and future work.

The experiments support a direct view of schedule design. Efficient schedules place effective allocation near the noise levels where posterior uncertainty changes fastest. The main remaining limitation is boundary behavior. Conditional-entropy-rate estimates can become singular near deterministic limits, so a more principled regularization could sharpen the profile estimate and improve endpoint accuracy. We did not test high-resolution regimes, where the profile may decompose across scales or semantic levels. Extending the allocation view to structured, blockwise, or multimodal corruption paths is a natural next step, since components may resolve uncertainty at different stages of the generative trajectory.

Impact Statement

This paper advances the methodological foundations of diffusion training by replacing manual noise-schedule search with a data-adaptive allocation principle grounded in information dynamics along the corruption path. Reducing trial-and-error schedule design can lower the compute and engineering burden of training diffusion models across new datasets, resolutions, and representations, accelerating research and broadening access. Improved efficiency can also lower the marginal cost of high-fidelity synthetic data, amplifying both beneficial applications and harmful misuse.

Acknowledgements

Gabriel Raya was funded by the Dutch Research Council (NWO) as part of the CERTIF-AI project (file number 17998).

References

Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and Van Den Berg, R. (2021). Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993.

Biroli, G., Bonnaire, T., De Bortoli, V., and Mézard, M. (2024). Dynamical regimes of diffusion models. *Nature Communications*, 15(1):9957.

Chen, T. (2023). On the importance of noise scheduling for diffusion models. *arXiv preprint arXiv:2301.10972*.

Chen, Y., Liang, C., Sui, H., Guo, R., Cheng, C., You, J., and Liu, G. (2026). Langflow: Continuous diffusion rivals discrete in language modeling. *arXiv preprint arXiv:2604.11748*.

DaSilva, L. F., Senan, S., Kribelbauer-Swietek, J. F., Patel, Z. M., Louis, L. K., Reddy, A. J., Gabbita, S., Rosen, J. D., Nussbaum, Z., Córdova, C. M. V., et al. (2026). Designing synthetic regulatory elements using the generative ai framework dna-diffusion. *Nature Genetics*, 58(1):180–194.

Dieleman, S., Sartran, L., Roshannai, A., Savinov, N., Ganin, Y., Richemond, P. H., Doucet, A., Strudel, R., Dyer, C., Durkan, C., et al. (2022). Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*.

Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al. (2024). Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*.

Esteves, C. and Makadia, A. (2026). Spectrally-guided diffusion noise schedules. *arXiv preprint arXiv:2603.19222*.

Franzese, G., Bounoua, M., and Michiardi, P. (2024). Minde: Mutual information neural diffusion estimation. In *International Conference on Learning Representations*, volume 2024, pages 16685–16716.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., et al. (2020). The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Guo, D., Shamai, S., and Verdú, S. (2005). Mutual information and minimum mean-square error in gaussian channels. *IEEE transactions on information theory*, 51(4):1261–1282.

Hang, T., Gu, S., Li, C., Bao, J., Chen, D., Hu, H., Geng, X., and Guo, B. (2023). Efficient diffusion training via min-snr weighting strategy. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7441–7451.

He, Z., Sun, T., Tang, Q., Wang, K., Huang, X.-J., and Qiu, X. (2023). Diffusionbert: Improving generative masked language models with diffusion models. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 4521–4534.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.

Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. J. (2022). Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646.

Hoogeboom, E., Heek, J., and Salimans, T. (2023). simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, pages 13213–13232. PMLR.

Kahouli, K., Hessmann, S. S. P., Müller, K.-R., Nakajima, S., Gugler, S., and Gebauer, N. W. A. (2024). Molecular relaxation by reverse diffusion with time step prediction. *Machine Learning: Science and Technology*, 5(3):035038.

Karras, T., Aittala, M., Aila, T., and Laine, S. (2022). Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577.

Karras, T., Aittala, M., Lehtinen, J., Hellsten, J., Aila, T., and Laine, S. (2024). Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24174–24184.

Karras, T., Laine, S., and Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410.

Kingma, D. and Gao, R. (2023). Understanding diffusion objectives as the elbo with simple data augmentation. *Advances in Neural Information Processing Systems*, 36:65484–65516.

Kingma, D., Salimans, T., Poole, B., and Ho, J. (2021). Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707.

Kong, X., Brekelmans, R., and Steeg, G. V. (2023). Information-theoretic diffusion. In *The Eleventh International Conference on Learning Representations*.

Kong, Z., Ping, W., Huang, J., Zhao, K., and Catanzaro, B. (2020). Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*.

Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario.

Lai, C.-H., Song, Y., Kim, D., Mitsufuji, Y., and Ermon, S. (2025). The principles of diffusion models. *arXiv preprint arXiv:2510.21890*.

Li, M. and Chen, S. (2024). Critical windows: non-asymptotic theory for feature emergence in diffusion models. *arXiv preprint arXiv:2403.01633*.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. (2023). Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*.

Nichol, A. Q. and Dhariwal, P. (2021). Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR.

Palomar, D. and Verdu, S. (2006). Gradient of mutual information in linear vector gaussian channels. *IEEE Transactions on Information Theory*, 52(1):141–154.

Pillutla, K., Swayamdipta, S., Zellers, R., Thickstun, J., Welleck, S., Choi, Y., and Harchaoui, Z. (2021). Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828.

Premkumar, A. (2026). Neural entropy. *Advances in Neural Information Processing Systems*, 38:21626–21669.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*.

Raya, G. and Ambrogioni, L. (2023). Spontaneous symmetry breaking in generative diffusion models. *Advances in Neural Information Processing Systems*, 36:66377–66389.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695.

Sahoo, S. S., Arriola, M., Schiff, Y., Gokaslan, A., Marroquin, E., Chiu, J. T., Rush, A., and Kuleshov, V. (2024). Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184.

Sclocchi, A., Favero, A., and Wyart, M. (2025). A phase transition in diffusion models reveals the hierarchical nature of data. *Proceedings of the National Academy of Sciences*, 122(1):e2408799121.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Shi, J., Han, K., Wang, Z., Doucet, A., and Titsias, M. (2024). Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37:103131–103167.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*.

Song, Y. and Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2020). Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Stancevic, D., Handke, F., and Ambrogioni, L. (2026). Entropic time schedulers for generative diffusion models. *Advances in Neural Information Processing Systems*, 38:44222–44252.

Stančević, D. and Ambrogioni, L. (2026). The information dynamics of generative diffusion. *Entropy*, 28(2).

Wang, C., Franzese, G., Gallo, M., Michiardi, P., et al. (2025). Information theoretic text-to-image alignment. In *International Conference on Learning Representations*, volume 2025, pages 50574–50610.

Wang, Y., Schiff, Y., Gokaslan, A., Pan, W., Wang, F., De Sa, C., and Kuleshov, V. (2023). Infodiffusion: Representation learning using information maximizing diffusion models. In *International conference on machine learning*, pages 36336–36354. PMLR.

Xu, Y., Tong, S., and Jaakkola, T. S. (2023). Stable target field for reduced variance score estimation in diffusion models. In *The Eleventh International Conference on Learning Representations*.

Yang, D., Zhang, Y., Yu, X., Hou, L., Tao, X., Wan, P., Qi, X., and Liao, R. (2026). Stable velocity: A variance perspective on flow matching. *arXiv preprint arXiv:2602.05435*.

Supplementary material

The supplement provides the derivations and implementation details underlying the allocation view, the information-profile target, and the online estimator used in the experiments.

- Appendix A rewrites diffusion and flow-matching objectives in a common x-prediction view, separating the sampling density from the loss weight and deriving the induced allocation $\phi_x = \pi w_x$ for the schedules used in the paper.
- Section B derives the conditional-entropy-rate profile for affine-Gaussian corruption paths from I–MMSE, and gives the finite-dataset Bayes quantities and two-point decision-window example used to interpret localized uncertainty resolution.
- Section C specifies the online estimator, sampler rebuilds, warm-up, smoothing, endpoint regularization, datasets, architectures, optimization settings, samplers, NFE conventions, guidance settings, metrics, and compute-to-target protocol.
- Section D provides allocation plots, profile galleries, empirical Bayes visualizations, qualitative samples, and consistency checks for the binned loss estimate.

A Noise schedules and weights in a common allocation view

Diffusion and flow-matching methods are often written in different path coordinates, prediction targets, and loss weights. Directly comparing their schedules therefore confounds how often path locations are sampled with how strongly their errors are weighted. For the affine-Gaussian paths used in our baselines, these objectives can be rewritten in a common x-prediction view indexed by relative noise or log-SNR. Following the design-space view of Karras et al. (2022) and the weighted log-SNR view of Kingma and Gao (2023), this appendix derives the induced allocation $\phi_x = \pi w_x$ for the fixed schedules used in our experiments.

A.1 Affine-Gaussian paths

For a linear forward SDE, the perturbation kernel can be written as

$$\mathbf{x}_t = s(t)\mathbf{x}_0 + s(t)\sigma(t)\boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{17}$$

with

$$s(t) = \exp\left(\int_0^t f(\xi)\, d\xi\right), \qquad \sigma(t)^2 = \int_0^t \frac{g(\xi)^2}{s(\xi)^2}\, d\xi. \tag{18}$$

Here $s(t)$ is the signal scale and $\sigma(t)$ is the relative noise scale in the standardized kernel notation of Karras et al. (2022). For allocation comparisons, the induced kernel is the relevant object. After a monotone reparameterization $t = t(u)$, we write the same path as

$$\mathbf{x}_u = a(u)\mathbf{x}_0 + b(u)\boldsymbol{\varepsilon}, \qquad a(u) = s(t(u)), \qquad b(u) = s(t(u))\sigma(t(u)), \qquad a(u), b(u) > 0. \tag{19}$$

The coordinate u indexes path locations. This affine-Gaussian form covers the standard VE, VP, and sub-VP perturbation kernels up to endpoint limits. EDM is included through the VE-style perturbation kernel, and our flow-matching baselines through affine-Gaussian conditional interpolants.

A.2 Standardized denoising coordinate

Factoring out the signal scale rewrites Equation (19) in the Karras-style standardized form

$$\mathbf{x}_u = a(u)\left(\mathbf{x}_0 + \frac{b(u)}{a(u)}\boldsymbol{\varepsilon}\right) = a(u)\big(\mathbf{x}_0 + \sigma(u)\boldsymbol{\varepsilon}\big), \tag{20}$$

with

$$\sigma(u) \coloneqq \frac{b(u)}{a(u)}, \qquad \gamma(u) \coloneqq \frac{a(u)^2}{b(u)^2} = \frac{1}{\sigma(u)^2}, \qquad \lambda(u) \coloneqq \log\gamma(u) = -2\log\sigma(u). \tag{21}$$

For allocation comparisons in the denoising coordinate, the deterministic scale $a(u)$ does not change the posterior inference problem at fixed u.

$$\bar{\mathbf{x}}_u \coloneqq \frac{\mathbf{x}_u}{a(u)} = \mathbf{x}_0 + \sigma(u)\boldsymbol{\varepsilon}. \tag{22}$$

Thus the denoising problem can be indexed by relative noise $\sigma(u)$, or equivalently by log-SNR $\lambda(u)$.

A schedule is usually specified in an implementation coordinate η, not directly in λ. If $\eta \sim \pi_\eta$ and $\lambda = \lambda(\eta)$ is monotone, the induced density over log-SNR is

$$\pi(\lambda) = \pi_\eta(\eta(\lambda))\left|\frac{d\eta}{d\lambda}\right|. \tag{23}$$

The schedule rule samples its native coordinate, while $\pi(\lambda)$ describes how often training visits denoising problems in the common coordinate.

In the x-prediction view, a weighted denoising objective has the form

$$\mathcal{L}_x = \int \pi(\lambda) w_x(\lambda)\, \mathbb{E}\Big[\|\mathbf{x}_0 - \hat{\mathbf{x}}_0(\bar{\mathbf{x}}_\lambda, \lambda)\|_2^2 \mid \lambda\Big]\, d\lambda. \tag{24}$$

The effective allocation is

$$\phi_x(\lambda) \coloneqq \pi(\lambda) w_x(\lambda). \tag{25}$$

The schedule determines how often a denoising problem is visited. The loss weight determines how strongly its error contributes. Their product is the allocation compared across schedules.

A.3 Continuous-time VDM objective and entropy rate

The continuous-time VDM objective of Kingma et al. (2021) provides a useful likelihood-based reference point. Let

$$\mathbf{z}_t = a(t)\mathbf{x}_0 + b(t)\boldsymbol{\varepsilon}, \qquad \gamma(t) = \frac{a(t)^2}{b(t)^2}, \qquad \gamma'(t) \le 0.$$

The diffusion term of the continuous-time bound can be written in x-prediction form as

$$\mathcal{L}_\infty(\theta) = -\frac{1}{2}\int_0^1 \gamma'(t)\mathbb{E}_{\mathbf{x}_0,\boldsymbol{\varepsilon}}\left[\|\mathbf{x}_0 - \hat{\mathbf{x}}_\theta(\mathbf{z}_t;t)\|_2^2\right] \mathrm{d}t. \tag{26}$$

The coefficient $-\frac{1}{2}\gamma'(t)$ is induced by the continuous-time ELBO and the SNR parameterization of the path. It is not an additional loss weight.

At the Bayes x-prediction denoiser,

$$\hat{\mathbf{x}}_{\theta^\star}(\mathbf{z}_t;t) = \mathbb{E}[\mathbf{x}_0 \mid \mathbf{z}_t],$$

the expected denoising loss equals

$$\mathrm{mmse}(t) = \mathbb{E}\left[\|\mathbf{x}_0 - \mathbb{E}[\mathbf{x}_0 \mid \mathbf{z}_t]\|_2^2\right].$$

After standardization,

$$\frac{\mathbf{z}_t}{b(t)} = \sqrt{\gamma(t)}\,\mathbf{x}_0 + \boldsymbol{\varepsilon}.$$

The Gaussian-channel I–MMSE identity gives

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{H}[\mathbf{x}_0 \mid \mathbf{z}_t] = -\frac{1}{2}\gamma'(t)\,\mathrm{mmse}(t). \tag{27}$$

Thus, at the Bayes denoiser, the continuous-ELBO integrand equals the conditional-entropy rate. For a learned denoiser, the integrand remains model-dependent because the denoising loss need not equal $\mathrm{mmse}(t)$.

A.4 Prediction-space conversions

Many objectives are written in ϵ-prediction form. From Equation (22),

$$\boldsymbol{\varepsilon} = \frac{\bar{\mathbf{x}}_u - \mathbf{x}_0}{\sigma(u)}. \tag{28}$$

Given an ϵ-prediction model $\hat{\boldsymbol{\varepsilon}}_\theta(\bar{\mathbf{x}}_u;u)$, define its induced x-reconstruction as

$$\hat{\mathbf{x}}_{\boldsymbol{\theta}}^{(\epsilon)}(\bar{\mathbf{x}}_u;u) \coloneqq \bar{\mathbf{x}}_u - \sigma(u)\hat{\boldsymbol{\varepsilon}}_\theta(\bar{\mathbf{x}}_u;u). \tag{29}$$

Then

$$\mathbf{x}_0 - \hat{\mathbf{x}}_{\boldsymbol{\theta}}^{(\epsilon)} = \sigma(u)\left(\hat{\boldsymbol{\varepsilon}}_\theta - \boldsymbol{\varepsilon}\right), \tag{30}$$

and therefore

$$\|\boldsymbol{\varepsilon} - \hat{\boldsymbol{\varepsilon}}_\theta\|_2^2 = \frac{1}{\sigma(u)^2}\left\|\mathbf{x}_0 - \hat{\mathbf{x}}_{\boldsymbol{\theta}}^{(\epsilon)}\right\|_2^2 = e^{\lambda(u)}\left\|\mathbf{x}_0 - \hat{\mathbf{x}}_{\boldsymbol{\theta}}^{(\epsilon)}\right\|_2^2. \tag{31}$$

An ϵ-prediction objective with allocation

$$\phi_\epsilon(\lambda) = \pi(\lambda)w_\epsilon(\lambda)$$

therefore induces, in the common x-reconstruction view,

$$\phi_x(\lambda) = e^\lambda\phi_\epsilon(\lambda) = e^\lambda\pi(\lambda)w_\epsilon(\lambda). \tag{32}$$

A.5 Common support and plotting coordinate

All allocation comparisons use the same finite log-SNR interval $[\lambda_{\min},\lambda_{\max}]$. If $\pi(\lambda)$ is the density on its full support, the truncated training density is

$$\tilde{\pi}(\lambda) = \frac{\pi(\lambda)\mathbf{1}\{\lambda \in [\lambda_{\min},\lambda_{\max}]\}}{\int_{\lambda_{\min}}^{\lambda_{\max}}\pi(\ell)\,d\ell}. \tag{33}$$

The corresponding unnormalized effective allocation is

$$\tilde{\phi}_x(\lambda) = \tilde{\pi}(\lambda)w_x(\lambda), \tag{34}$$

and allocation plots normalize $\tilde{\phi}_x$ on the common support.

For plots shown on a logarithmic σ-axis, using $\lambda = -2\log\sigma$, the density with respect to σ is

$$\psi_x(\sigma) = \phi_x(\lambda(\sigma))\left|\frac{d\lambda}{d\sigma}\right| = \frac{2}{\sigma}\phi_x(-2\log\sigma). \tag{35}$$

The density displayed per unit $\log\sigma$ is proportional to $\sigma\psi_x(\sigma)$, equivalently to $\phi_x(-2\log\sigma)$.

A.6 Baseline allocations

We instantiate the schedule densities used in the main experiments in the common log-SNR coordinate λ. All expressions below are densities with respect to λ, shown up to normalization on the common support. When a schedule is used under a fixed x-prediction objective, its effective allocation is obtained by multiplying the listed $\pi(\lambda)$ by the experiment's fixed x-space weight $w_x(\lambda)$.

EDM. EDM (Karras et al., 2022) samples

$$\log \sigma \sim \mathcal{N}(P_{\text{mean}}, P_{\text{std}}^2). \tag{36}$$

Since $\lambda = -2 \log \sigma$, this induces

$$\pi_{\text{EDM}}(\lambda) = \mathcal{N}\big(\lambda;\, -2P_{\text{mean}}, (2P_{\text{std}})^2\big)\,. \tag{37}$$

With the EDM x-space denoising weight

$$w_x^{\text{EDM}}(\sigma) = \frac{\sigma^2 + \sigma_{\text{data}}^2}{\sigma^2 \sigma_{\text{data}}^2}, \tag{38}$$

we obtain, up to a constant independent of λ,

$$w_x^{\text{EDM}}(\lambda) \propto 1 + \sigma_{\text{data}}^2 e^{\lambda}, \qquad \phi_x^{\text{EDM}}(\lambda) \propto \pi_{\text{EDM}}(\lambda)\big(1 + \sigma_{\text{data}}^2 e^{\lambda}\big). \tag{39}$$

Log-uniform in σ. A log-uniform schedule satisfies

$$\pi_{\text{LU}}(\sigma) = \frac{1}{\sigma \log(\sigma_{\max}/\sigma_{\min})}. \tag{40}$$

Since $\lambda = -2 \log \sigma$, the induced density $\pi_{\text{LU}}(\lambda)$ is uniform on

$$[-2 \log \sigma_{\max}, -2 \log \sigma_{\min}].$$

Therefore, under a fixed x-prediction objective,

$$\phi_x^{\text{LU}}(\lambda) = \pi_{\text{LU}}(\lambda) w_x(\lambda) \propto w_x(\lambda).$$

In our experiments using constant x-space loss weight, this gives a uniform effective allocation in λ. Log-uniform sampling therefore serves as the allocation-neutral reference.

Cosine / iDDPM. For the continuous cosine limit of Nichol and Dhariwal (2021),

$$\bar{\alpha}(t) = \cos^2\left(\frac{\pi t}{2}\right), \qquad \sigma(t) = \tan\left(\frac{\pi t}{2}\right), \qquad t \in (0, 1). \tag{41}$$

Uniform sampling in t induces

$$\pi_{\cos}(\lambda) \propto \left|\frac{dt}{d\lambda}\right| \propto \operatorname{sech}\left(\frac{\lambda}{2}\right). \tag{42}$$

If cosine is used with unit-weight ϵ-prediction, Equation (32) gives

$$\phi_x^{\cos}(\lambda) \propto e^{\lambda} \operatorname{sech}(\lambda/2). \tag{43}$$

If cosine is used only as a schedule under a fixed x-prediction objective, its allocation is instead

$$\phi_x^{\cos}(\lambda) \propto \pi_{\cos}(\lambda) w_x(\lambda).$$

Flow matching with the OT path. For the OT interpolation (Lipman et al., 2023),

$$\mathbf{x}_t = (1 - t)\mathbf{x}_0 + t\boldsymbol{\varepsilon}, \qquad t \in (0, 1), \tag{44}$$

the relative-noise and log-SNR coordinates are

$$\sigma(t) = \frac{t}{1 - t}, \qquad \lambda(t) = \log \sigma(t)^{-2} = 2 \log \frac{1 - t}{t}. \tag{45}$$

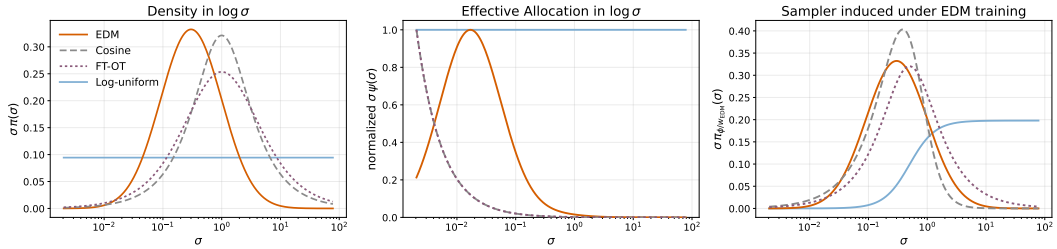


Figure 10: **Baseline schedules induce different effective allocations.** Curves are expressed in the common x-prediction view on the shared log-SNR support. The plotted allocation combines the sampling density and the x-space loss weight, $\phi_x(\lambda) = \pi(\lambda)w_x(\lambda)$.

Table 3: **Baseline allocations in the common x-prediction view.** Densities are with respect to $\lambda = \log \mathrm{SNR}$ and are shown up to normalization on the common support. Symbolic $w_x(\lambda)$ denotes the fixed x-space loss weight of the corresponding experiment.

Baseline	Sampling rule	$\pi(\lambda)$	$\phi_x(\lambda)$
EDM	$\log \sigma \sim \mathcal{N}(P_{\mathrm{mean}}, P_{\mathrm{std}}^2)$	$\mathcal{N}(\lambda; -2P_{\mathrm{mean}}, (2P_{\mathrm{std}})^2)$	$\propto \pi_{\mathrm{EDM}}(\lambda)(1+\sigma_{\mathrm{data}}^2 e^{\lambda})$
Log-uniform	$\log \sigma \sim \mathcal{U}$	const.	$\propto w_x(\lambda)$
Cosine / iDDPM	$t \sim \mathcal{U}(0,1)$	$\propto \mathrm{sech}(\lambda/2)$	$\propto \pi_{\cos}(\lambda)w_x(\lambda)$
FM-OT	$t \sim \mathcal{U}(0,1)$	$\propto \mathrm{sech}^2(\lambda/4)$	$\propto e^{\lambda/2}$

Uniform sampling in t induces

$$\pi_{\mathrm{OT}}(\lambda) \propto \left|\frac{dt}{d\lambda}\right| \propto \mathrm{sech}^2(\lambda/4). \tag{46}$$

For the velocity target $v = \boldsymbol{\varepsilon} - \mathbf{x}_0$,

$$\mathbf{x}_0 = \mathbf{x}_t - tv, \qquad \hat{\mathbf{x}}_0 = \mathbf{x}_t - t\hat{v}. \tag{47}$$

Thus

$$\|v - \hat{v}\|_2^2 = \frac{1}{t^2}\|\mathbf{x}_0 - \hat{\mathbf{x}}_0\|_2^2. \tag{48}$$

The unit-weight velocity objective therefore induces

$$w_x^{\text{FM-OT}}(\lambda) = \frac{1}{t(\lambda)^2} = \left(1 + e^{\lambda/2}\right)^2. \tag{49}$$

Combining this weight with $\pi_{\mathrm{OT}}(\lambda)$, and using

$$\mathrm{sech}^2(\lambda/4) \propto \frac{e^{\lambda/2}}{(1+e^{\lambda/2})^2},$$

gives

$$\phi_x^{\text{FM-OT}}(\lambda) = \pi_{\mathrm{OT}}(\lambda)w_x^{\text{FM-OT}}(\lambda) \propto e^{\lambda/2}. \tag{50}$$

B Conditional-entropy rates and I–MMSE

The information profile used by InfoNoise is the conditional-entropy rate along an affine-Gaussian corruption path. This appendix derives the rate from I–MMSE, gives the VE and log-SNR forms used in the paper, and records finite-dataset Bayes formulas together with the two-point decision-window example.

B.1 Affine-Gaussian paths and entropy rate

Consider the affine-Gaussian corruption path

$$\mathbf{x}_u = a(u)\mathbf{x}_0 + b(u)\boldsymbol{\varepsilon}, \qquad \mathbf{x}_0 \sim p_{\text{data}}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d), \qquad a(u), b(u) > 0. \tag{51}$$

For fixed u, dividing by the noise scale is an invertible transformation of the observation,

$$\tilde{\mathbf{x}}_u := \frac{\mathbf{x}_u}{b(u)} = \sqrt{\gamma(u)}\,\mathbf{x}_0 + \boldsymbol{\varepsilon}, \qquad \gamma(u) := \frac{a(u)^2}{b(u)^2}. \tag{52}$$

Thus $\mathbf{x}_u$ and $\tilde{\mathbf{x}}_u$ define the same posterior over $\mathbf{x}_0$, and

$$\mathrm{H}[\mathbf{x}_0 \mid \mathbf{x}_u] = \mathrm{H}[\mathbf{x}_0 \mid \tilde{\mathbf{x}}_u]. \tag{53}$$

Define the Bayes denoising error at SNR γ by

$$\mathrm{mmse}(\gamma) := \mathbb{E}\left[\|\mathbf{x}_0 - \mathbb{E}[\mathbf{x}_0 \mid \sqrt{\gamma}\mathbf{x}_0 + \boldsymbol{\varepsilon}]\|_2^2\right]. \tag{54}$$

The Gaussian-channel I–MMSE identity (Guo et al., 2005; Palomar and Verdu, 2006) gives

$$\frac{\partial}{\partial\gamma} I(\mathbf{x}_0; \sqrt{\gamma}\mathbf{x}_0 + \boldsymbol{\varepsilon}) = \frac{1}{2}\,\mathrm{mmse}(\gamma). \tag{55}$$

Since

$$\mathrm{H}[\mathbf{x}_0 \mid \mathbf{x}_u] = \mathrm{H}[\mathbf{x}_0] - I(\mathbf{x}_0; \tilde{\mathbf{x}}_u),$$

the chain rule yields

$$\frac{d}{du}\mathrm{H}[\mathbf{x}_0 \mid \mathbf{x}_u] = -\frac{1}{2}\,\gamma'(u)\,\mathrm{mmse}(\gamma(u)). \tag{56}$$

The conditional-entropy rate is therefore determined by the Bayes-optimal denoising error at the current SNR and by how quickly the path changes SNR.

For an arbitrary monotone coordinate, we define the pathwise information profile as the magnitude of this derivative,

$$\rho_u^\star(u) \propto \left|\frac{d}{du}\mathrm{H}[\mathbf{x}_0 \mid \mathbf{x}_u]\right| = \frac{1}{2}\,|\gamma'(u)|\,\mathrm{mmse}(\gamma(u)). \tag{57}$$

Large values mark path locations where posterior uncertainty about $\mathbf{x}_0$ changes most rapidly in the chosen coordinate.

B.2 VE and log-SNR forms

For the VE channel,

$$\mathbf{x}_\sigma = \mathbf{x}_0 + \sigma\boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d), \tag{58}$$

we have $a(\sigma) \equiv 1$, $b(\sigma) = \sigma$, and $\gamma(\sigma) = \sigma^{-2}$. Substituting into Equation (56) gives

$$\frac{d}{d\sigma}\mathrm{H}[\mathbf{x}_0 \mid \mathbf{x}_\sigma] = \frac{\mathrm{mmse}(\sigma)}{\sigma^3}, \tag{59}$$

where

$$\mathrm{mmse}(\sigma) := \mathbb{E}\left[\|\mathbf{x}_0 - \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_\sigma]\|_2^2\right]. \tag{60}$$

Equivalently, with log-SNR $\lambda = \log\gamma = -2\log\sigma$,

$$-\frac{d}{d\lambda}\mathrm{H}[\mathbf{x}_0 \mid \mathbf{x}_\sigma] = \frac{1}{2}\,\gamma\,\mathrm{mmse}(\gamma) = \frac{1}{2}\,\frac{\mathrm{mmse}(\sigma)}{\sigma^2}. \tag{61}$$

When training is parameterized by VE noise scale σ, Equation (59) is the profile estimated online by INFONOISE.

The profiles in Equations (59) and (61) are densities with respect to different path coordinates. Under the change of variables $\lambda = -2\log\sigma$, the density transforms by the Jacobian $|d\sigma/d\lambda| = \sigma/2$. Alignment with a training allocation is meaningful only after both quantities are expressed in the same coordinate.

B.3 Common path examples

The identity in Equation (56) applies to any affine-Gaussian path.

VE. For $a(u) \equiv 1$, $b(u) = \sigma(u)$,

$$\gamma(u) = \sigma(u)^{-2}, \qquad \frac{d}{du}\mathrm{H}[\mathbf{x}_0 \mid \mathbf{x}_u] = \frac{\sigma'(u)}{\sigma(u)^3}\ \mathrm{mmse}(\gamma(u)). \tag{62}$$

VP. For

$$\mathbf{x}_u = \alpha(u)\mathbf{x}_0 + \sqrt{1-\alpha(u)^2}\,\boldsymbol{\varepsilon},$$

we have

$$\gamma(u) = \frac{\alpha(u)^2}{1-\alpha(u)^2},$$

and therefore

$$\frac{d}{du}\mathrm{H}[\mathbf{x}_0 \mid \mathbf{x}_u] = -\frac{\alpha(u)\alpha'(u)}{(1-\alpha(u)^2)^2}\,\mathrm{mmse}(\gamma(u)). \tag{63}$$

Affine-Gaussian flow matching. For the linear Gaussian interpolant

$$\mathbf{x}_u = (1-u)\mathbf{x}_0 + u\boldsymbol{\varepsilon},$$

we have

$$a(u) = 1-u, \qquad b(u) = u, \qquad \gamma(u) = \frac{(1-u)^2}{u^2}.$$

Thus

$$\frac{d}{du}\mathrm{H}[\mathbf{x}_0 \mid \mathbf{x}_u] = \frac{1-u}{u^3}\ \mathrm{mmse}(\gamma(u)). \tag{64}$$

B.4 Empirical Bayes formulas for finite datasets

Under an empirical prior, the Bayes denoiser and posterior covariance have closed-form finite-sum expressions. Given $\{x_i\}_{i=1}^N \subset \mathbb{R}^d$, let

$$p_{\mathrm{data}}(x_0) = \frac{1}{N}\sum_{i=1}^N \delta(x_0 - x_i). \tag{65}$$

Under the VE channel, the noisy marginal is the Gaussian mixture

$$p(x;\sigma) = \frac{1}{N}\sum_{i=1}^N \mathcal{N}(x; x_i, \sigma^2\mathbf{I}_d). \tag{66}$$

Bayes' rule gives posterior weights

$$w_i(x;\sigma) = \frac{\mathcal{N}(x; x_i, \sigma^2\mathbf{I}_d)}{\sum_{j=1}^N \mathcal{N}(x; x_j, \sigma^2\mathbf{I}_d)}, \qquad \sum_{i=1}^N w_i(x;\sigma) = 1, \tag{67}$$

and the Bayes denoiser is

$$\hat{x}^\star(x;\sigma) = \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_\sigma = x] = \sum_{i=1}^N w_i(x;\sigma)x_i. \tag{68}$$

The posterior covariance is

$$\mathrm{Cov}(\mathbf{x}_0 \mid \mathbf{x}_\sigma = x) = \sum_{i=1}^N w_i(x;\sigma)\big(x_i - \hat{x}^\star(x;\sigma)\big)\big(x_i - \hat{x}^\star(x;\sigma)\big)^\top, \tag{69}$$

so

$$\mathrm{mmse}(\sigma) = \mathbb{E}_{x\sim p(\cdot;\sigma)}\left[\mathrm{tr}\,\mathrm{Cov}(\mathbf{x}_0 \mid \mathbf{x}_\sigma = x)\right]. \tag{70}$$

Substituting this quantity into Equation (59) gives the empirical Bayes entropy-rate profile used for offline analysis. These profiles are diagnostics only and are not provided to INFONOISE during training.

The same empirical posterior gives the score through Tweedie's formula,

$$\nabla_x \log p(x;\sigma) = \frac{\hat{x}^\star(x;\sigma) - x}{\sigma^2}. \tag{71}$$

Stationary points of the noisy marginal therefore satisfy

$$\nabla_x \log p(x;\sigma) = 0 \quad \Longleftrightarrow \quad x = \hat{x}^\star(x;\sigma). \tag{72}$$

B.5 Two-point decision window

The information-window behavior can be seen analytically in a symmetric two-point distribution. Let

$$p_{\text{data}}(x_0) = \frac{1}{2}\delta(x_0 - a) + \frac{1}{2}\delta(x_0 + a), \qquad a > 0. \tag{73}$$

The noisy marginal under the VE channel is

$$p(x;\sigma) = \frac{1}{2}\mathcal{N}(x;a,\sigma^2) + \frac{1}{2}\mathcal{N}(x;-a,\sigma^2). \tag{74}$$

The posterior weight on $+a$ is

$$w_+(x;\sigma) = \frac{1}{1+\exp\left(-\frac{2ax}{\sigma^2}\right)}, \qquad w_-(x;\sigma) = 1 - w_+(x;\sigma), \tag{75}$$

so the Bayes denoiser is

$$\hat{x}^\star(x;\sigma) = a\big(w_+(x;\sigma) - w_-(x;\sigma)\big) = a\tanh\left(\frac{ax}{\sigma^2}\right). \tag{76}$$

Since $x_0 \in \{\pm a\}$, the posterior variance is

$$\mathrm{Var}(x_0 \mid x_\sigma = x) = a^2 - \hat{x}^\star(x;\sigma)^2 = a^2\,\mathrm{sech}^2\left(\frac{ax}{\sigma^2}\right), \tag{77}$$

and

$$\mathrm{mmse}(\sigma) = \mathbb{E}_{x\sim p(\cdot;\sigma)}\left[a^2\,\mathrm{sech}^2\left(\frac{ax}{\sigma^2}\right)\right]. \tag{78}$$

The VE entropy-rate profile is therefore $\mathrm{mmse}(\sigma)/\sigma^3$.

Using Equation (71), the score is

$$\frac{\partial}{\partial x}\log p(x;\sigma) = \frac{a\tanh\left(\frac{ax}{\sigma^2}\right) - x}{\sigma^2}. \tag{79}$$

Stationary points satisfy

$$x = a\tanh\left(\frac{ax}{\sigma^2}\right). \tag{80}$$

Nonzero fixed-point branches appear when the slope of the right-hand side at the origin exceeds one,

$$\frac{a^2}{\sigma^2} > 1 \qquad \Longleftrightarrow \qquad \sigma < \sigma_c \coloneqq a. \tag{81}$$

Equivalently,

$$\frac{\partial^2}{\partial x^2}\log p(0;\sigma) = \frac{a^2 - \sigma^2}{\sigma^4}, \tag{82}$$

so $x = 0$ changes from a local maximum of the noisy marginal to a local minimum as σ crosses σ_c. The noisy marginal changes from unimodal to bimodal, and the Bayes denoiser stops averaging over both modes and begins committing to one branch. This transition is the analytic version of the information window in the main text. At high noise, observations carry little information about which atom generated the sample. At very low noise, the posterior is already concentrated. The entropy-rate profile peaks in the intermediate region where posterior uncertainty collapses most rapidly.

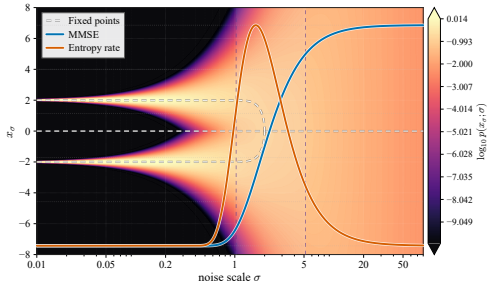


Figure 11: **Entropy rate localizes the decision window.** MMSE measures remaining uncertainty; entropy rate marks where that uncertainty changes fastest. Fixed-point branches show the same transition geometrically.

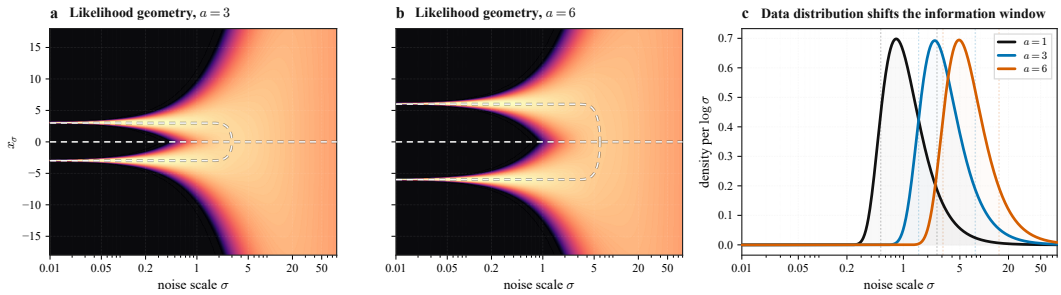


Figure 12: **The data distribution sets the information window.** Under the same VE path, changing the atom scale a in $x_0 \in \{-a, +a\}$ shifts where posterior uncertainty is resolved. **(a,b)** Likelihood geometry for $a = 1$ and $a = 6$. Bayes-denoiser fixed points move along the noise axis as data scale changes. **(c)** The entropy-rate density shifts with a, showing that the informative region is induced by the data distribution under the corruption path, not the noise coordinate alone.

Data dependence of the decision window. The two-point example also shows that the information window is set by the data distribution under the corruption path, not by the noise coordinate alone. In the prior $x_0 \in \{-a, +a\}$, the only data scale is the atom scale a. Writing $x = ay$ and $\sigma = ar$, the posterior weight becomes

$$w_+(ay; ar) = \frac{1}{1 + \exp\left(-\frac{2y}{r^2}\right)},$$

so posterior uncertainty depends on the relative noise level $r = \sigma/a$. Consequently, the Bayes denoising error has the scaling form

$$\mathrm{mmse}_a(\sigma) = a^2\, m\left(\frac{\sigma}{a}\right)$$

for a dimensionless function m. The VE entropy-rate density is

$$\rho_a(\sigma) \propto \frac{\mathrm{mmse}_a(\sigma)}{\sigma^3} = \frac{1}{a}\frac{m(\sigma/a)}{(\sigma/a)^3}.$$

Thus changing the data scale shifts the information window along the same VE path. Figure 12 shows this shift explicitly.

After normalization as a density over σ, the profile has the same shape in the relative coordinate σ/a, so its peak shifts linearly with the data scale a. The fixed-point transition occurs at $\sigma_c = a$, while the entropy-rate maximum occurs at a constant multiple of a. Thus the schedule coordinate fixes how noise is indexed, but the data distribution determines where uncertainty is resolved.

B.6 Regularization of the estimated information profile

INFONOISE constructs its training sampler from an online estimate of the information profile. Before normalization, this estimate can contain low-noise boundary structure that is not the interior uncertainty-resolution region we want to allocate training around. If left untreated, normalization can place excessive mass near the smallest noise levels and reduce coverage of the region where posterior uncertainty changes. We therefore apply a smooth low-noise gate to the estimated profile before converting it into a sampling density. The gate regularizes the estimate; it is not an additional schedule family.

Let $r(\sigma) \geq 0$ denote the binned and smoothed entropy-rate estimate at a sampler rebuild. The low-noise boundary has different origins for continuous and discrete endpoints. For continuous data, $\mathrm{H}[\mathbf{x}_0 \mid \mathbf{x}_\sigma]$ is a differential entropy. As $\sigma \to 0$, the posterior concentrates with covariance of order σ^2, so

$$\mathrm{H}[\mathbf{x}_0 \mid \mathbf{x}_\sigma] = d \log \sigma + O(1).$$

Under standard regularity conditions, $\mathrm{mmse}(\sigma) = \Theta(\sigma^2)$. The VE entropy-rate identity gives

$$\frac{\mathrm{d}}{\mathrm{d}\sigma}\mathrm{H}[\mathbf{x}_0 \mid \mathbf{x}_\sigma] = \frac{\mathrm{mmse}(\sigma)}{\sigma^3} = \Theta(\sigma^{-1}) \qquad (\sigma \to 0). \tag{83}$$

This contribution is finite on the truncated training interval $[\sigma_{\min}, \sigma_{\max}]$ with $\sigma_{\min} > 0$, but it can dominate normalization near the lower endpoint. For discrete endpoints, conditional entropy remains bounded as $\sigma \to 0$; empirically, the estimated profile can instead contain an extended low-noise power-law segment,

$$r(\sigma) \propto \sigma^{\alpha}, \qquad \log r(\sigma) = \text{const} + \alpha \log \sigma. \tag{84}$$

The mechanisms differ, but the practical issue is the same: endpoint structure in the raw estimate can pull allocation away from the interior information-bearing region.

We suppress this boundary structure with the multiplicative gate

$$g_{c,n}(\sigma) = \frac{\sigma^n}{\sigma^n + c^n}, \qquad c > 0, \quad n \geq 2. \tag{85}$$

For $\sigma \gg c$, $g_{c,n}(\sigma) \approx 1$; for $\sigma \ll c$, $g_{c,n}(\sigma) \approx (\sigma/c)^n$. Combining the gate with the continuous low-noise scaling in Equation (83) gives

$$\left(\frac{\mathrm{d}}{\mathrm{d}\sigma}\mathrm{H}[\mathbf{x}_0 \mid \mathbf{x}_\sigma]\right) g_{c,n}(\sigma) = \Theta(\sigma^{n-1}) \qquad (\sigma \to 0), \tag{86}$$

so the gated profile vanishes at the boundary for $n \geq 2$. For discrete endpoints, the same gate attenuates the empirical low-noise power-law segment. We use $n = 3$ in all experiments.

The regularized allocation target is

$$\rho(\sigma) = \frac{r(\sigma) g_{c,n}(\sigma)}{\int_{\sigma_{\min}}^{\sigma_{\max}} r(\tilde{\sigma}) g_{c,n}(\tilde{\sigma})\, d\tilde{\sigma}}. \tag{87}$$

The sampler is then built from this target using the fixed training weight, as in Equation (15),

$$\pi(\sigma) \propto \rho(\sigma)/w(\sigma),$$

followed by normalization and inverse-CDF sampling. When w is constant, the sampler density is proportional to ρ.

Gate pivot. The gate form and exponent are fixed across experiments, with $n = 3$. The pivot c is a boundary regularization parameter, not a schedule parameter selected by validation performance. For continuous image experiments, c is computed from the current profile using an onset-of-information rule. Let $r(\sigma)$ denote the ungated online estimate and define

$$\bar{r}(\sigma) = \frac{r(\sigma)}{\max_{\tilde{\sigma}} r(\tilde{\sigma})}.$$

Scanning from high to low noise, we set c at the first persistent crossing of a fixed threshold p,

$$c = \sup\left\{\sigma \,\middle|\, \bar{r}(\sigma') < p \text{ for all } \sigma' \geq \sigma\right\}.$$

We use $p = 0.002$. On CIFAR-10 this gives $c \approx 0.153$, rounded to $c = 0.15$.

For discrete shifted-regime experiments, the low-noise boundary appears as an extended power-law segment in log–log coordinates. In the reported DNA experiments we use the fixed gate pivot $c = 0.2$, with the same exponent $n = 3$. This pivot is motivated by the observed boundary structure and is kept fixed across runs. The gate regularizes the estimated profile before normalization; it is not an additional schedule family or a validation-tuned sampling distribution.

C Experimental and reproducibility details

The experiments isolate training allocation. Within each controlled comparison, the architecture, denoising objective, loss weighting, optimizer, EMA, sampler, inference budget, and evaluation protocol are fixed; only the training noise distribution changes. Image experiments follow the EDM setup of Karras et al. (2022). Discrete endpoint experiments keep binary or tokenized data at the clean endpoint, but train with a continuous VE Gaussian corruption process.

The main text defines the comparison protocol, compute-to-target metric, and use of offline diagnostic profiles. This appendix provides dataset-specific architectures, training settings, samplers, metrics, and compute resources for reproducibility.

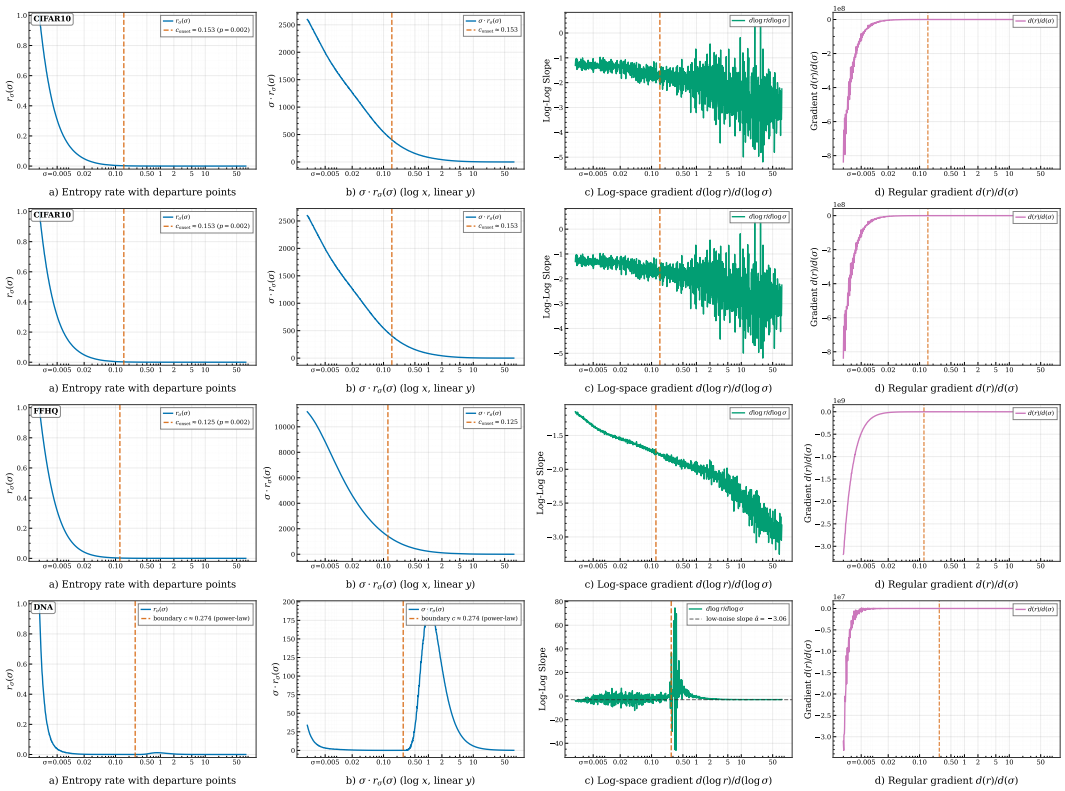


Figure 13: **Low-noise boundary structure motivates gated entropy-rate profiles.** Continuous domains show low-noise entropy-rate growth, while discrete domains show extended low-noise power-law structure in the raw estimate. Vertical markers indicate the detected pivot c, the upper edge of the boundary-dominated region. The gate suppresses endpoint structure before normalization while preserving the interior information-bearing region.

C.1 INFONOISE sampler implementation

The online estimator is maintained on a fixed grid in the training coordinate. Each batch records the sampled noise level and the unweighted x-space denoising error ℓ. Losses are binned in $\log \sigma$, smoothed across refreshes, multiplied by the VE path factor, gated near the low-noise endpoint using Section B.6, and normalized to obtain the target profile $\hat{\rho}(\sigma)$. The sampler is rebuilt from $\pi(\sigma) \propto \hat{\rho}(\sigma)/w(\sigma)$, followed by normalization and inverse-CDF sampling. This keeps the loss weight fixed while adapting the sampling density so that the induced allocation follows the estimated profile.

Before the first reliable estimate is available, INFONOISE samples from a fixed warm-up prior π_0. After the first refresh, all subsequent sampler rebuilds use the online profile estimate. Warm-up length, smoothing, minimum bin counts, refresh cadence, and endpoint gating are fixed within each domain.

C.2 Image experiments

For CIFAR-10 and FFHQ 64×64, we use the EDM image setup of Karras et al. (2022) without architecture or optimizer retuning. Images are represented in $[-1, 1]$. The denoiser uses the EDM preconditioning coefficients $c_{\text{in}}, c_{\text{skip}}, c_{\text{out}}, c_{\text{noise}}$ with $\sigma_{\text{data}} = 0.5$, $\sigma_{\min} = 0.002$, and $\sigma_{\max} = 80$. The EDM baseline samples $\log \sigma$ from the original log-normal training distribution. INFONOISE keeps the denoiser, preconditioning, loss weight, optimizer, EMA, augmentations, and sampler fixed, and changes only the training distribution over σ after warm-up.

Sampling uses the deterministic EDM probability-flow sampler with a second-order Heun solver and the $\rho = 7$ noise grid. All schedules use the same solver, noise range, checkpoint budget, and FID

protocol. We use NFE $= 35$ for CIFAR-10 and NFE $= 79$ for FFHQ 64×64. FID is computed from $50{,}000$ generated samples against the standard real-image statistics.

C.3 Discrete endpoint experiments

The shifted representation and sequence experiments keep the clean endpoint discrete while using a continuous VE Gaussian corruption process. For binary endpoints $\mathbf{x}_0 \in \{0,1\}^D$, the network outputs Bernoulli logits at each position, which are converted to probabilities

$$\hat{\mathbf{x}}_0(\mathbf{x}_\sigma;\sigma) = \mathrm{sigmoid}(\ell_\theta(\mathbf{x}_\sigma,\sigma)).$$

Training uses probability-space denoising with the same fixed loss weight within each schedule comparison. For sampling, the probability denoiser gives the VE score estimate

$$\nabla_{\mathbf{x}_\sigma}\widehat{\log p(\mathbf{x}_\sigma;\sigma)} = \frac{\hat{\mathbf{x}}_0(\mathbf{x}_\sigma;\sigma)-\mathbf{x}_\sigma}{\sigma^2},$$

which is integrated with the same second-order Heun solver across schedules. Samples are decoded by thresholding the final probabilities at 0.5.

Within each dataset, schedule variants use the same architecture, optimizer, objective, sampler, and evaluation protocol. EDM transfer uses the image-domain EDM log-normal sampler with $P_{\mathrm{mean}} = -1.2$, $P_{\mathrm{std}} = 1.2$, and w_{EDM}. Log-uniform sampling uses $w(\sigma) \equiv 1$. InfoNoise also uses $w(\sigma) \equiv 1$, but replaces the fixed sampler after warm-up with the online information-profile estimate. Self-conditioning and classifier-free guidance are used only in the text experiments.

Binarized CIFAR-10. CIFAR-10 images are converted to flattened Gray-coded RGB bitstreams. We use $45{,}000$ training images, $5{,}000$ validation images, and the standard test split. Each $32\times32\times3$ image is represented with 24 bits per pixel, giving $D = 24{,}576$. Bits are flattened in raster order. The model is a multiscale 1D U-Net transformer with base width 64, time embedding dimension 128, head dimension 32, and 8 levels. The hierarchy uses downsampling factors $(2,2,2,2,2,2,2)$, width multipliers $(1,1,2,2,4,6,8,8)$, and blocks per level $(1,1,1,2,2,3,3,4)$. Local attention with window size 128 is used at the first three levels and global attention at coarser levels; shifted windows, RoPE, Fourier positional features, and SwiGLU are enabled. The noisy input is centered at 0.5 and scaled by $(\sigma^2+\sigma_{\mathrm{data}}^2)^{-1/2}$.

All binarized CIFAR-10 runs use batch size 64, 1000 epochs, EMA decay 0.9997, AdamW, learning rate 2×10^{-4}, $(\beta_1,\beta_2) = (0.9, 0.99)$, weight decay 10^{-2}, gradient clipping at 1.0, cosine learning-rate decay with warm-up, mixed precision where supported, PyTorch compilation, and FlashAttention-compatible kernels when available. Sampling uses 128 Heun steps, requiring 255 NFE, with $\sigma_{\max}^{\mathrm{eval}} = 20.0$ and $\sigma_{\mathrm{decode}} = 0.1$. We report FID on $50{,}000$ generated samples.

OpenWebText-2 bitstreams. For OpenWebText-2, we train a byte-level BPE tokenizer with vocabulary size $32{,}768$. Tokens are ordered by a semantic traversal of skip-gram Word2Vec embeddings and encoded with Gray code. The resulting 15-bit token code is expanded to 21 bits using SECDED-style error correction. Each example contains 256 tokens, giving $D = 5376$. We build a 10^9-token cache and split it deterministically into train, validation, and test partitions with fractions $0.99/0.005/0.005$. Training uses conditional prefix completion. Prefix length is sampled uniformly from 0 to 128 tokens. Prefix bits remain clean, suffix bits are corrupted, and the loss is evaluated only on the suffix. Classifier-free guidance is trained by dropping the prefix with probability 0.2 and replacing it with the null value 0.5. The model is a flat transformer over 21-bit patches with content embedding dimension 64, trunk dimension 512, 16 transformer blocks, 8 attention heads, RoPE, AdaLN-Zero conditioning, local absolute positional side features, SwiGLU feed-forward layers, and self-conditioning probability 0.5. A skip-MLP head combines patch-level context with fine per-bit features and outputs Bernoulli logits for every bit. All OpenWebText-2 runs use batch size 256, 66 epochs, approximately 10^6 optimization steps, and EMA decay 0.9999. Optimizer and numerical settings match the bitstream CIFAR-10 runs. Sampling uses 64 Heun steps, requiring 127 NFE, and stops at $\sigma = 0.185$, where the expected bit MSE is approximately 10^{-12}. At inference, conditional and unconditional probability denoisers are combined before conversion to the VE score. We evaluate guidance scales $\{0.0, 1.5, 2.0\}$ and report 2.0 in the main text. Generated bitstreams are thresholded, decoded by reversing ECC and semantic Gray coding, and converted to text with the trained tokenizer. We report MAUVE on $10{,}240$ generated test samples using `gpt2-large` features, maximum text length 256, and micro-batch size 512.

C.4 Metrics, inference budgets, and compute

Metrics are computed with the same inference sampler and NFE within each dataset. Image and bitstream-image experiments report FID from $50{,}000$ generated samples. DNA experiments report Sei-FID. Text experiments report MAUVE on $10{,}240$ generated samples from the test split using `gpt2-large` features. The NFE reported in the main table is the number of denoiser evaluations used by the evaluation sampler.

All reported experiments were run on NVIDIA H100 GPUs. CIFAR-10 and FFHQ 64×64 used 8 GPUs per run; all other experiments used 4 GPUs on the same cluster. Main-text compute budgets are reported in processed examples. Wall-clock ranges are hardware-dependent references: CIFAR-10 took $\sim$2d 00h–2d 02h to 200kimg, FFHQ 64×64 took $\sim$2d 09h–2d 13h to 200kimg, binarized CIFAR-10 took $\sim$7h–9h to 200kimg.

C.5 Existing assets and licenses

We use public datasets, model components, and evaluation tools. The original sources are cited below, and license or terms information is reported from the corresponding source when available. We do not redistribute any dataset as a new asset.

Table 4: **Existing assets used in the experiments.** License and terms entries follow the corresponding source documentation where available.

Asset	Use in this paper	License / terms
CIFAR-10 (Krizhevsky and Hinton, 2009)	Image and bitstream experiments	Public benchmark dataset; original Toronto page describes the dataset but does not state an explicit license
FFHQ (Karras et al., 2019)	FFHQ 64×64 image experiments	Images retain their original Flickr licenses; dataset metadata and tooling are released under the terms stated by the FFHQ source
OpenWebText-2 (Gao et al., 2020)	Text bitstream experiments	MIT license, as stated by the EleutherAI OpenWebText2 repository
EDM code and setup (Karras et al., 2022)	Image baseline, preconditioning, loss weighting, and sampler	Official NVIDIA research code release; repository includes a license file and directs licensing inquiries to NVIDIA Research Licensing
MAUVE (Pillutla et al., 2021)	Text evaluation metric	GPLv3, as stated by the `mauve-text` package metadata
GPT-2-large (Radford et al., 2019)	Feature extractor for MAUVE	Modified MIT License, as stated by the model release

D Allocation diagnostics and qualitative results

This appendix collects diagnostics that connect the online estimator to the allocation behavior used in the main experiments. We first show how unweighted denoising losses are converted into an information-profile estimate and then into an effective allocation. We then provide dataset-level profile galleries, qualitative samples at matched checkpoints, and a consistency argument for the binned loss estimate used during sampler rebuilds.

D.1 Consistency of the binned loss estimate

Fix a denoiser $\hat{x}_\theta$, a sampler π, and a bin $B_j \subset \mathcal{U}$ with nonzero sampler mass. Define

$$L_\theta(u) = \mathbb{E}\big[\|x_0 - \hat{x}_\theta(x_u; u)\|_2^2 \mid u\big] .$$

If the conditional loss variance is finite, the empirical average of losses whose sampled noise levels fall in B_j converges almost surely to

$$\mathbb{E}_\pi[L_\theta(u) \mid u \in B_j] .$$

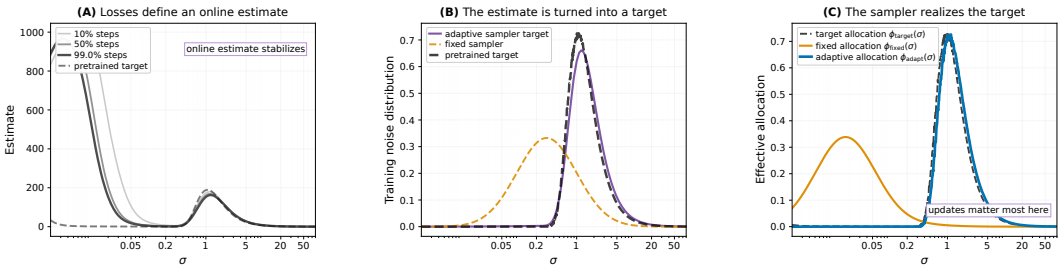
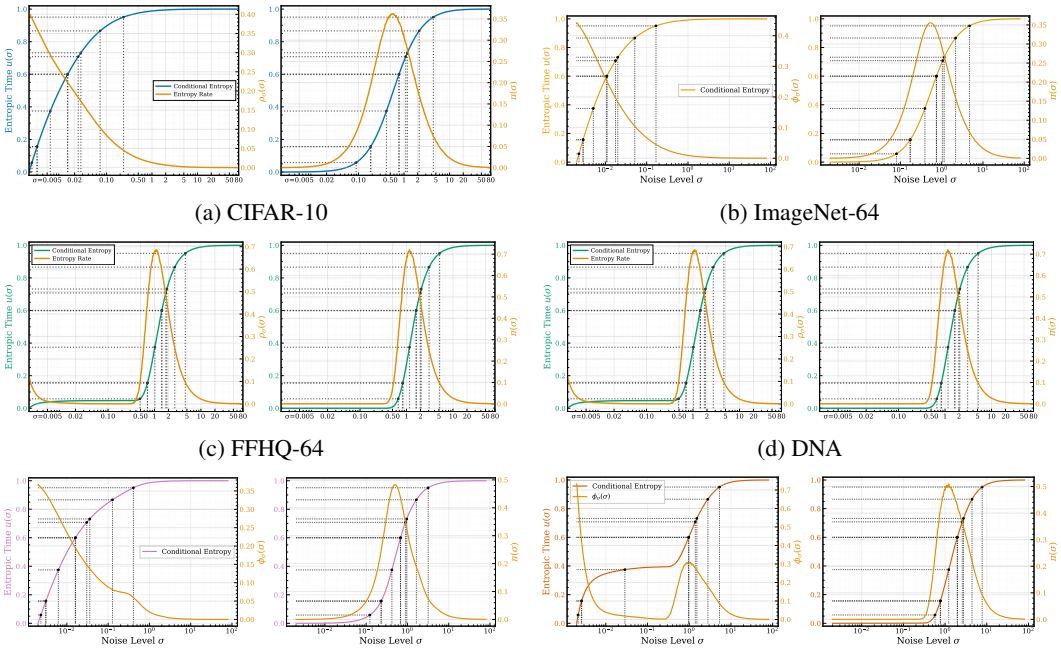


Figure 14: **INFONOISE turns online loss statistics into effective allocation.** **(A)** Unweighted denoising losses measure the model's x-space error across noise levels. **(B)** The path factor converts these losses into an online information-profile estimate, shown alongside a fixed reference sampler and a post hoc diagnostic profile. **(C)** Rebuilding the sampler changes the induced effective allocation while the objective and loss weighting remain fixed.



(a) CIFAR-10

(b) ImageNet-64

(c) FFHQ-64

(d) DNA

(e) Binarized MNIST

(f) MNIST

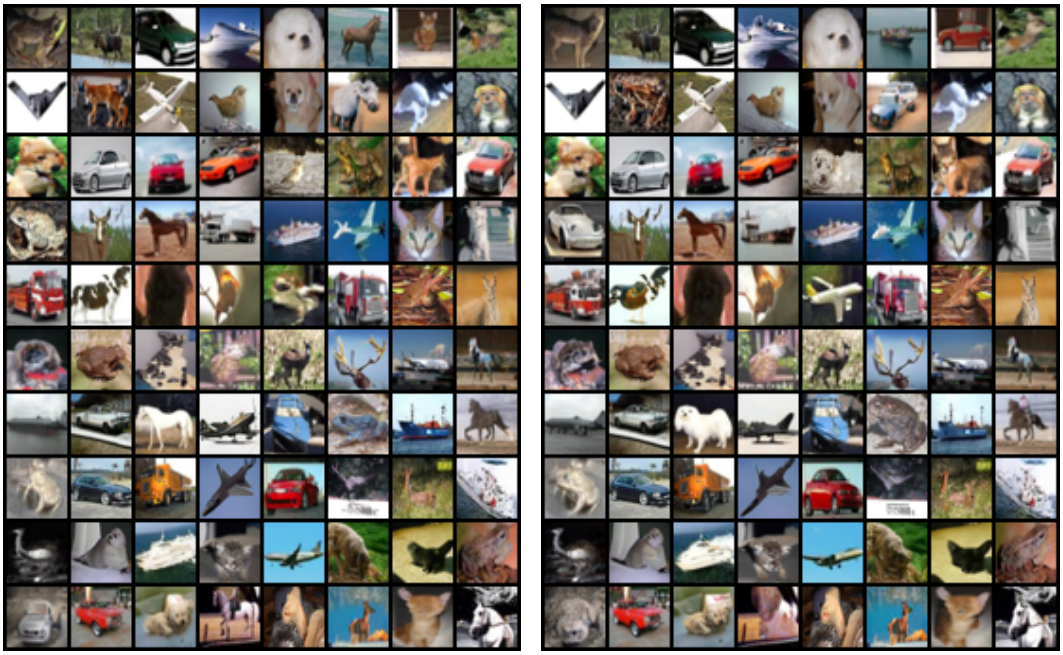
Figure 15: **INFONOISE estimates dataset-specific information profiles.** Each panel shows the normalized online profile estimate and the inverse-CDF sampler induced by INFONOISE. The sampler $\pi(\sigma)$, together with the fixed per-noise weight $w(\sigma)$, determines the realized allocation $\phi(\sigma) = \pi(\sigma)w(\sigma)$. The profiles show that the informative region shifts across image, discrete, and sequence regimes.

As the grid is refined and $L_\theta(u)$ is continuous, this conditional average converges to $L_\theta(u_j)$ for any representative point $u_j \in B_j$. If the denoiser is Bayes-optimal,

$$\hat{x}_\theta(x_u; u) = \mathbb{E}[x_0 \mid x_u],$$

then $L_\theta(u) = \mathrm{mmse}(\gamma(u))$. Multiplying by $\frac{1}{2}|\gamma'(u)|$ gives the conditional-entropy-rate profile up to normalization. In the online algorithm, this binned statistic is used as a plug-in estimate at each sampler refresh; smoothing, minimum bin counts, warm-up, and endpoint gating keep the finite-sample estimate stable while the model and sampler evolve.

Together, these diagnostics support the main claim that INFONOISE adapts the training distribution through an online estimate of where denoising uncertainty is resolved, rather than through post hoc profile access or a changed objective.



(a) **CIFAR-10 @100k** INFONOISE, Heun, NFE= 35 (b) **CIFAR-10 @100k** EDM, Heun, NFE= 35

(c) **FFHQ-64 @160k** INFONOISE, Heun, NFE= 79 (d) **FFHQ-64 @160k** EDM, Heun, NFE= 79

Figure 16: **Qualitative samples at matched training checkpoints.** INFONOISE and EDM are evaluated with the same solver, inference grid, and NFE. Checkpoints are matched by processed examples.