PRIVACY ACCOUNTING FOR EMPIRICAL DENOISERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models have emerged as the foundation of modern generative systems, yet their high memorization capacity raises privacy concerns. While differentially private (DP) training provides formal guarantees, it comes at a cost in sample quality. In this work, we study a complementary question: *how much privacy loss is incurred during the sampling process itself?* Directly analyzing modern neural denoisers is challenging because their sensitivity to a single training example is generally intractable. To overcome this challenge, we construct and analyze a *clipped privacy-accountable empirical denoiser*, a training-free estimator of the Bayes-optimal denoiser, whose per-step sensitivity can be computed explicitly, allowing each denoising step to be a Gaussian mechanism. This perspective enables privacy accounting under Gaussian DP when each training example contributes to each denoising step, and under f-DP trade-off functions when each generated sample uses a single randomly subsampled block of the training data. We report every (ε, δ) guarantee as a certified upper bound from a privacy loss distribution accountant, and use a cumulant-based Edgeworth approximation as a fast diagnostic, characterizing where it is reliable and where it becomes anti-conservative. We also analyze the privacy loss along the denoising trajectory and show that it grows steeply near the critical window in which semantic content is formed. This observation motivates hybrid samplers that use the private empirical denoiser through the formative steps where it remains a faithful proxy for the neural denoiser, then switch to a public neural denoiser for the remaining refinement. Our results on CIFAR-10 and CelebA-64 show that the privacy of diffusion models depends not only on the training procedure, but also on sampling-time design choices such as subsampling, stopping time, and hybrid denoising.

1 INTRODUCTION

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) are now central to modern generative systems, but their ability to reproduce training examples raises serious privacy concerns (Wu et al., 2023; Carlini et al., 2023; Tang et al., 2024; Hu & Pang, 2023; Duan et al., 2023; Matsumoto et al., 2023). Post-hoc filters and memorization-aware model changes can reduce visible leakage, but they do not give attack-agnostic privacy guarantees. Differentially private (DP) training gives such guarantees, but it answers a different question: it bounds what the weights reveal, while users and regulators see only generated samples. Sampling-time leakage is a separate surface that persists even when the denoiser is trained non-privately, whether or not DP training is practical at scale. This raises a complementary question: *how much privacy loss is incurred by the sampling process itself?*

Answering this requires treating the sampler, rather than the training algorithm, as the mechanism to be analyzed. The obstacle is that a neural denoiser has intractable data sensitivity, so its denoising steps admit no explicit privacy accounting. To make the problem analyzable, we privatize an *empirical denoiser* (Niedoba et al., 2025) whose sensitivity can be computed directly. Since each reverse diffusion step already injects Gaussian noise, the empirical denoising update can be interpreted as a Gaussian mechanism, allowing us to track the sampling trajectory with Gaussian Differential Privacy (GDP) (Dong et al., 2022).

However, the GDP composition using the full batch of training data yields a poor privacy level. Denoising with subsampled data at each denoising step can improve the privacy level. However, this hurts the diffusion sampling, as every newly drawn subset of data injects extra stochastic drift into the sampling trajectory, which prevents coherent semantics from forming across the trajectory. To

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

this end, we propose *output-wise subsampling*, subsample a block from the dataset once and reuse it for the entire denoising trajectory to produce one sample. This stabilizes the denoising direction and achieves a decent privacy level.

When converting f-DP to (ϵ, δ)-DP, we encounter overly pessimistic conversion, especially at small δ, when using the CLT-type f-DP accounting that assumes the CDF of the standardized test statistic under the composition is simply Gaussian. We report all guarantees as certified upper bounds from a PLD accountant (Doroshenko et al., 2022), and study a cumulant-based *Edgeworth approximation* (Zheng et al., 2020) as a fast estimate and measure where it is accurate and where it understates the certified value.

Our contributions are summarized below:

1. We privatize a clipped empirical denoiser with explicit per-step sensitivity of $2C/|\mathcal{D}|$. Diffusion sampling then becomes a composition of Gaussian mechanisms, with no model trained on the private data.
2. We derive privacy guarantees for full-batch and output-wise subsampled diffusion sampling using GDP and f-DP trade-off functions, and report certified (ϵ, δ) upper bounds from a PLD accountant. We use the Edgeworth approximation as a diagnostic where cumulant checks pass, and characterize where it fails. Subsampling amplification applies once per reused block, rather than once per step.
3. We show that the per-step privacy cost grows along the trajectory relative to the critical window, measure where the empirical denoiser is a window-local semantic proxy for a neural denoiser, and use both to place the private-to-public handoff, evaluated on CIFAR-10 and CelebA-64.

2 Background

2.1 Denoising Diffusion Models

Diffusion models are built upon the forward and reverse processes (Ho et al., 2020; Song et al., 2020). The forward perturbation describes how the noise is gradually corrupting data sample $\mathbf{x}_0$, defined as $\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \epsilon$, where ϵ is a standard Gaussian variable, α_t and σ_t are prescribed time-dependent mean and variance coefficients, respectively. For simplicity, we describe the diffusion process from EDM (Karras et al., 2022), but our framework is adaptable to any diffusion processes such as DDPM (Ho et al., 2020) or Flow Matching (FM) (Lipman et al., 2022). In EDM, the forward diffusion process is simply defined by $\mathbf{x}_t = \mathbf{x}_0 + t\epsilon$, i.e., $\alpha_t \equiv 1$ and $\sigma_t = t$, and the reverse denoising process is computed by

$$\mathbf{x}_{t-\Delta t} = \left(1 - 2\frac{\Delta t}{t}\right)\mathbf{x}_t + 2\frac{\Delta t}{t}\mathbb{E}[\mathbf{x}|\mathbf{x}_t] + \sqrt{2t\Delta t}\epsilon. \tag{1}$$

The term $\mathbb{E}[\mathbf{x}|\mathbf{x}_t]$, called a Bayes optimal denoiser, is the conditional expectation of the clean data given the noisy observation $\mathbf{x}_t$. In practice, a neural denoiser $D(\mathbf{x}_t, t; \theta(\mathcal{D}))$ approximates this quantity, where $\theta(\mathcal{D})$ are parameters learned from dataset $\mathcal{D}$. By rolling out eq. 1 up to time zero, diffusion model gradually synthesizes a sample.

2.2 Differential Privacy and Gaussian Mechanism

Differential Privacy (DP) provides a formal framework to limit the influence of any single record on the output of a randomized algorithm. Intuitively, whether or not an individual's data is present, the distribution of outputs should remain (almost) unchanged. This view naturally extends to diffusion sampling, which is itself a randomized mechanism.

A fundamental building block is the *Gaussian mechanism*: given a function $h : \mathcal{D} \mapsto \mathbb{R}^p$, the mechanism releases

$$\mathcal{M}_{\text{Gauss}}(\mathcal{D}; h) = h(\mathcal{D}) + n, \qquad n \sim \mathcal{N}(0, \sigma^2 \Delta_h^2 \mathbf{I}), \tag{2}$$

where Δ_h is the global sensitivity—the maximum change in h between neighboring datasets. By calibrating the noise to Δ_h, the Gaussian mechanism guarantees DP (see Appendix H.1 for formal definitions).

For analyzing multi-step procedures such as diffusion sampling, we adopt *Gaussian Differential Privacy (GDP)* (Dong et al., 2022). GDP tightly characterizes the privacy loss of Gaussian mechanisms and, crucially, admits an exact composition rule:

$$\mu = \sqrt{\sum_i \mu_i^2}, \tag{3}$$

where μ_i is the GDP parameter of the i-th step. For example, in the case of eq. 2 with $n \sim \mathcal{N}(0, \sigma^2 \Delta_{\mathbf{h}}^2 \mathbf{I})$, the Gaussian mechanism corresponds to μ_i-GDP with $\mu_i = \frac{1}{\sigma}$. This makes GDP especially well-suited for diffusion processes, which inject Gaussian noise at every denoising step.

Two further facts are important for our analysis. First, privacy amplification by subsampling can further reduce the effective μ. Second, GDP parameters can be translated into (ϵ, δ) guarantees when needed for regulatory reporting; we provide details in Appendix H.2. Together, these tools allow us to connect the denoising process directly to established DP accounting methods.

2.3 Privacy Loss Distributions

(ϵ, δ)-DP is equivalently characterized by the hockey-stick divergence: a mechanism is (ϵ, δ)-DP iff $H_{e^\epsilon}(M(\mathcal{D}) \,\|\, M(\mathcal{D}')) \leq \delta$ for all neighboring $\mathcal{D}, \mathcal{D}'$, where $H_\gamma(P\|Q) = \int [P(x) - \gamma Q(x)]_+ \, dx$. The *privacy loss distribution* (PLD) is the law of the log-likelihood ratio $\log \frac{P(x)}{Q(x)}$ under P; the hockey-stick divergence, and hence $\delta(\epsilon)$, is a functional of it, and the PLD of a composition is the convolution of the individual PLDs. Numerical PLD accountants evaluate this convolution on a discretized support (Gopi et al., 2021; Doroshenko et al., 2022). Discretization can be performed *pessimistically*, overestimating H_{e^ϵ} for every ϵ and hence yielding a valid (ϵ, δ) upper bound, or *optimistically*, underestimating it. We use the pessimistic estimate for all reported guarantees, and the optimistic estimate to quantify how much of the reported ϵ is discretization slack rather than true privacy loss. This distinction is what we mean by *certified* throughout: the reported value is an upper bound by construction, not an approximation of unknown sign.

3 Privacy Analysis of Denoising Process with Empirical Denoiser

3.1 Single-Step Denoising is a GDP

The Bayes denoiser in eq. 1 is the conditional expectation $\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t] = p_t(\mathbf{x}_t)^{-1} \int \mathbf{x}_0 \, p(\mathbf{x}_t|\mathbf{x}_0) \, p(\mathbf{x}_0) \, d\mathbf{x}_0$, where $p(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \mathbf{x}_0, t^2 I)$ is the closed-form EDM forward transition density and $p_t(\mathbf{x}_t) = \int p(\mathbf{x}_t \mid \mathbf{x}_0) \, p(\mathbf{x}_0) \, d\mathbf{x}_0$ is the marginal of the noisy state. Real-world samplers approximate this Bayes denoiser with a neural denoiser $D(\mathbf{x}_t, t; \theta(\mathcal{D}))$, but the sensitivity of D to changing one training example is not tractable, so it admits no direct privacy accounting. We therefore analyze the empirical denoiser, which replaces $p(\mathbf{x}_0)$ with the empirical measure over $\mathcal{D}$ and uses the empirical marginal $\widehat{p}_t(\mathbf{x}_t) = \frac{1}{|\mathcal{D}|} \sum_{\mathbf{x}_0 \in \mathcal{D}} p(\mathbf{x}_t \mid \mathbf{x}_0)$ as its normalizer. Unlike the transition kernel, $\widehat{p}_t$ is data-dependent and cannot enter a private mechanism without its own sensitivity accounting.

$$\hat{\mathbb{E}}[\mathbf{x}|\mathbf{x}_t; \mathcal{D}] = \frac{1}{\widehat{p}_t(\mathbf{x}_t)|\mathcal{D}|} \sum_{\mathbf{x}_0 \sim \mathcal{D}} \mathbf{x}_0 p(\mathbf{x}_t|\mathbf{x}_0). \tag{4}$$

Substituting eq. 4 into eq. 1 gives

$$\mathbf{x}_{t-\Delta t} = \Big(1 - \tfrac{2\Delta t}{t}\Big)\mathbf{x}_t + \tfrac{2\Delta t}{t} \underbrace{\tfrac{1}{p_t(\mathbf{x}_t)|\mathcal{D}|} \textstyle\sum_{\mathbf{x}_0 \in \mathcal{D}} \mathbf{x}_0 p(\mathbf{x}_t|\mathbf{x}_0)}_{\text{data-dependent}} + \sqrt{2t\Delta t}\, \epsilon, \tag{5}$$

so the only term depending on $\mathcal{D}$ is the empirical average, and the noise $\sqrt{2t\Delta t}\, \epsilon$ already injected by the sampler can serve as the noise of a Gaussian mechanism once this average has bounded sensitivity. Let $\mathbf{v}(\mathbf{x}_0; \mathbf{x}_t) = \mathbf{x}_0 p(\mathbf{x}_t|\mathbf{x}_0)$ and let $\bar{\mathbf{v}}(\mathbf{x}_0; \mathbf{x}_t) = \mathbf{v}(\mathbf{x}_0; \mathbf{x}_t) / \max(1, \|\mathbf{v}(\mathbf{x}_0; \mathbf{x}_t)\|/C)$ be its clipped version, so that $\|\bar{\mathbf{v}}\| \leq C$. Computing $\widehat{p}_t(\mathbf{x}_t)$ privately would require an additional data-dependent pass, so we replace it with a public constant β ($\beta = 1$ in experiments). This yields a deliberately

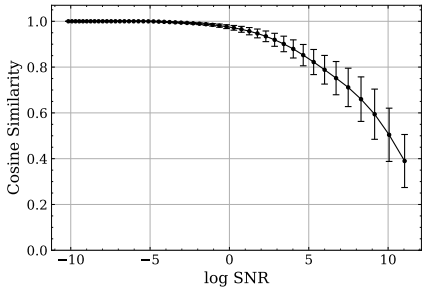


Figure 1: Cosine similarity between the denoising directions, i.e., $\mathrm{Cos}(\mathbf{x}_t - \hat{\mathbb{E}}, \mathbf{x}_t - D)$, by empirical and neural denoisers. Here, $\mathbf{x}_t - \hat{\mathbb{E}}$ denotes the actual denoising direction induced by the update rule in eq. 1. Results are reported along the log-SNR axis (Kingma et al., 2021), where $\mathrm{SNR} = 1/4t^2$, providing a unified noise-scale representation across diffusion formulations (e.g., EDM, DDPM, FM).

unnormalized empirical mechanism. $\beta = 1$ is a design choice, not a claim that $\widehat{p}_t(\mathbf{x}_t) \approx 1$, and β is a genuine privacy–utility parameter that we ablate in Appendix E. Substituting the clipped average into eq. 1 gives

$$\mathbf{x}_{t-\Delta t} = \Big(1 - \tfrac{2\Delta t}{t}\Big)\mathbf{x}_t + \tfrac{2\Delta t}{t}\,\tfrac{1}{\beta|\mathcal{D}|}\textstyle\sum_{\mathbf{x}_0\in\mathcal{D}} \bar{\mathbf{v}}(\mathbf{x}_0;\mathbf{x}_t) + \sqrt{2t\Delta t}\,\epsilon,$$

and the public invertible rescaling by $\beta t/(2\Delta t)$ shows that the data-dependent part of one step is the Gaussian mechanism

$$\mathcal{M}_{\mathrm{Gauss}}(\mathcal{D};\mathbf{x}_t) = \frac{1}{|\mathcal{D}|}\sum_{\mathbf{x}_0\in\mathcal{D}} \bar{\mathbf{v}}(\mathbf{x}_0;\mathbf{x}_t) + \beta\sqrt{\frac{t^3}{2\Delta t}}\,\epsilon. \tag{6}$$

Under replace-one adjacency ($\mathcal{D}, \mathcal{D}'$ differing in a single record), the clipped average has sensitivity at most $2C/|\mathcal{D}|$, so eq. 6 is μ_t-GDP with $\mu_t = \frac{2C}{\beta|\mathcal{D}|}\sqrt{2\Delta t/t^3}$. This bound is conservative in the sense that it treats the empirical denoising direction as the private release at every step and accounts for the sampling noise already injected by the diffusion process.

3.2 Multi-Step Denoising is also a GDP

For discretized timesteps t_i with increments Δt_i, the i-th empirical denoising step has $\mu_{t_i} = \frac{2C}{\beta|\mathcal{D}|}\sqrt{2\Delta t_i/t_i^3}$. By exact GDP composition, denoising from t_m to t_n is μ-GDP with $\mu = (\sum_{i=n}^{m}\mu_{t_i}^2)^{1/2}$. The parameter μ has a direct testing interpretation: larger μ means it is easier to distinguish neighboring datasets from the generated output. Privacy loss is therefore far from uniform along the trajectory. Under the EDM schedule $\Delta t_i \approx c\,t_i^{1-1/\rho}$, so $\mu_{t_i} \propto t_i^{-1-1/(2\rho)}$, which is roughly $1/t_i$ for $\rho = 7$ (Figure 5). The last low-noise steps therefore dominate the sum. This suggests a hybrid sampler: run the private empirical denoiser through the early and middle steps, then hand off to a public denoiser, whose steps are post-processing and add no privacy cost (Section 6).

3.3 Bridging Empirical and Neural Denoisers

A central limitation is that real diffusion samplers use neural denoisers, while our formal accounting uses an empirical denoiser. Directly accounting for neural denoisers remains open because the sensitivity of a trained network to one training example is not tractable. We therefore use the empirical denoiser as an analyzable proxy and check where this proxy is informative.

First, Figure 1 shows that empirical and neural denoising directions are nearly aligned on CIFAR-10 in the earlier stages of the denoising process. Second, recent work on critical windows (Li & Chen, 2024; Biroli et al., 2024; Sclocchi et al., 2024; Georgiev et al., 2023; Raya & Ambrogioni, 2023) suggests that coarse semantic identity is determined early and then remains stable, so the formative part of the trajectory overlaps with the regime where the two denoisers are most similar. Together, these checks do not prove a neural-denoiser DP guarantee, but they justify using the empirical accountant to study where sampling leakage occurs.

Cosine similarity captures only the denoising *direction*; the per-step magnitude gap $\|\widehat{\mathbb{E}} - D\|^2$ is reported in Appendix F, where it peaks inside the mid-noise window, but as Table 1 shows, does not survive to the released sample. We therefore restrict the proxy claim to directional agreement over the regime our private block occupies.

Table 1: Endpoint distance ($1 - \cos$) after substituting the empirical denoiser inside one window (swap) and after re-drawing only that window's noise with the neural denoiser kept (ref.). rel. = swap/ref.

		CLIP (Radford et al., 2021)			DINO (Oquab et al., 2024)		
Window	t range	swap	ref.	rel.	swap	ref.	rel.
high-noise	19.15–7.47	0.0097	0.0917	**0.11**×	0.0717	0.5792	**0.12**×
mid-noise	6.09–1.98	0.0772	0.1000	**0.77**×	0.5175	0.6203	**0.83**×
low-noise	1.54–0.38	0.1127	0.0687	1.64×	0.6910	0.4431	1.56×
min-noise	0.14–0.018	0.2147	0.0241	8.90×	0.8758	0.1376	6.37×

Trajectory-level agreement. The per-step divergence above bounds how far the two kernels can differ at a single step, but not how much of that difference survives to the released sample. We therefore substitute the empirical denoiser for the neural denoiser within one window at a time, with the initial noise and all stochastic increments held fixed, and measure the embedding distance ($1-\cos$) of the resulting sample to the all-neural reference trajectory. As a descriptive scale, we also keep the neural denoiser and re-draw only the noise increments inside that window. Table 1 reports both distances over 2048 paired trajectories, and Appendix F adds absolute similarities, paired-bootstrap CIs, PSNR, SSIM, and label agreement (Efron & Tibshirani, 1993; Wang et al., 2004). In the high- and mid-noise windows substitution moves the endpoint less than re-drawing that window's noise (mid-noise: $0.77\times$ CLIP, $0.83\times$ DINO, reference minus swap distance $+0.0228$ in CLIP, 95% CI $[+0.0207, +0.0248]$). In the mid-noise window endpoint similarity stays high in CLIP (0.923) but falls in DINO (0.483), so coarse semantics are preserved while instance-level structure is not. In the low-noise window the ordering reverses (CLIP difference -0.0441, 95% CI $[-0.0459, -0.0422]$), consistent with the t^{-3} growth of the per-step divergence. The min-noise window is always handled by the public denoiser, while the low-noise window is private in our 65% configuration (Section 6.3).

Taken together, these measurements support a claim: through the mid-noise window, about the first half of generation, the empirical denoiser is a window-local semantic proxy for the neural denoiser. They do not establish a DP guarantee for a neural denoiser by itself, which remains open.

4 Privacy Analysis of Denoising Process with Subsampled Data

Section 3 analyzes the empirical denoiser when every record in $\mathcal{D}$ contributes to every private denoising step. This full-batch accountant is tight, but it is expensive in terms of privacy: each generated output repeatedly touches every record, and privacy loss grows quickly when many outputs are released. Subsampling reduces this exposure because a record that is not selected contributes no privacy loss. Throughout, each generated output draws a minibatch $\mathcal{S} \subseteq \mathcal{D}$ of fixed size B uniformly without replacement, so a fixed record is included with probability $q_{\mathcal{B}} = B/n$, and neighboring datasets differ by replacing one record. The subsampled Gaussian mechanism is $\mathcal{M}_{\mathrm{SubGauss}}(\mathcal{D}; \mathbf{x}_t) = |\mathcal{S}|^{-1} \sum_{\mathbf{x}_0 \in \mathcal{S}} \bar{\mathbf{v}}(\mathbf{x}_0; \mathbf{x}_t) + \beta\sqrt{t^3/(2\Delta t)}\,\epsilon$.

The important design choice is how often we resample $\mathcal{S}$. Resampling at every denoising step for every output gives more independent amplification events, but it also changes the empirical denoising direction at every step. Small minibatches, which is better for privacy, inject extra stochastic drift into the reverse trajectory: the sample is repeatedly pulled toward different private subsets, and late low-noise steps are especially sensitive to this instability, as shown in Figure 2. This can harm sample quality even though the privacy accountant looks attractive. We therefore employ *output-wise subsampling*: sample once for a generated output, reuse them through a empirical denoising block, then account for amplification once at the block level. This keeps the denoising direction stable while still limiting how often any record is accessed.

We state both accountants because they correspond to different mechanisms. Applying per-step amplification to output-wise subsampling would be incorrect: privacy amplification is tied to independent subsampling events, not to the number of denoising steps. If the same minibatch is reused over a block $\mathcal{B}$, the included record accumulates the full block-level Gaussian parameter $\bar{\mu}_{\mathcal{B}} = (\sum_{t \in \mathcal{B}} \mu_t^2)^{1/2}$ before amplification is applied. If a fresh minibatch is drawn at every step, each μ_t is amplified separately.

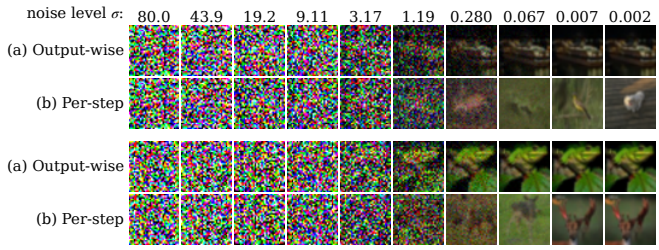


Figure 2: Output-wise vs. per-step subsampling, $B = 25$, identical initial noise. Each row is one sample; columns show decreasing noise levels. Under output-wise subsampling (a) the subsample is reused, so each row progressively denoises the same emerging image. Under per-step subsampling (b) a fresh subsample is drawn each step, pulling the trajectory toward a different subset: semantic content changes between columns and never settles. This instability motivates block reuse.

For full-batch Gaussian mechanisms, GDP composition is exact. With subsampling, each block's trade-off curve is a mixture rather than a Gaussian curve, so exact GDP composition no longer applies, and we compose numerically with a pessimistic PLD accountant (Section 2.3). The GDP central limit theorem (Dong et al., 2022) fits poorly here because the μ_t are highly heterogeneous, and its Berry-Esseen error is additive and too coarse for small δ (Table 11). As a fast estimate, we also compute a degree-two Edgeworth approximation (Zheng et al., 2020), which adds skewness and kurtosis corrections from the first four cumulants (eq. 25, Appendix D). It is an approximation and matches the certified PLD value to within 1% for $\varepsilon \leq 2$ but understates it for larger ε, by up to 15% in our runs (Appendix G).

Proposition 4.1 (Output-wise subsampling)**.** *Consider a sampler that, for each of N_{out} generated outputs, performs private subsampling once and reuses them for an empirical denoising block $\mathcal{B}$; all later public denoising is post-processing. Let $q_{\mathcal{B}}$ be the block-level inclusion probability of a fixed record and $\bar{\mu}_{\mathcal{B}} = (\sum_{t\in\mathcal{B}} \mu_t^2)^{1/2}$. Define $h_{\mathcal{B}} = q_{\mathcal{B}} G_{\bar{\mu}_{\mathcal{B}}} + (1 - q_{\mathcal{B}})\,\mathrm{Id}$ and*

$$g_{\text{out}} = \bigotimes_{j=1}^{N_{\text{out}}} h_{\mathcal{B}}, \qquad f_{\text{out}} = \min\{g_{\text{out}}, g_{\text{out}}^{-1}\}^{**}. \tag{7}$$

Then the joint release of the N_{out} generated outputs is f_{out}-DP. Equivalently, for every $\varepsilon \geq 0$ it is $(\varepsilon, \delta_{\text{out}}(\varepsilon))$-DP with

$$\delta_{\text{out}}(\varepsilon) = \max_{\text{dir}\in\{P\to Q, Q\to P\}} \left[Pr_Q(S_{\text{out}}^{\text{dir}} > \varepsilon) - e^{\varepsilon} Pr_P(S_{\text{out}}^{\text{dir}} > \varepsilon) \right]_+, \tag{8}$$

where $S_{\text{out}} = \sum_{j=1}^{N_{\text{out}}} L_{\mathcal{B}}(Z_j)$, $L_{\mathcal{B}}(z) = \log(1 - q_{\mathcal{B}} + q_{\mathcal{B}} \exp(\bar{\mu}_{\mathcal{B}} z - \bar{\mu}_{\mathcal{B}}^2/2))$, and $\bigotimes$ indicates a tensor product. The probabilities are taken under the two neighboring-data hypotheses. We evaluate $\delta_{\text{out}}(\varepsilon)$ with a pessimistic PLD accountant, which gives the certified values we report, and use the Edgeworth expansion only as an estimate (Appendix G).

Algorithm 1 instantiates this: the subsample is drawn once per output (line 3) and reused across the block, so amplification is applied once to $\bar{\mu}_{\mathcal{B}}$ rather than per step. In practice, each output enters the PLD accountant (Section 2.3) as one subsampled Gaussian mechanism with noise multiplier $1/\bar{\mu}_{\mathcal{B}}$ and sampling probability $q_{\mathcal{B}}$, whose two neighboring directions give exactly $h_{\mathcal{B}}$ and $h_{\mathcal{B}}^{-1}$. Composing N_{out} copies gives $\delta_{\text{out}}(\varepsilon)$, and Appendix G lists every step and setting. The proof is in Appendix C, which also gives the accountant for per-step resampling (Proposition C.1). That accounting is valid only when the implementation truly resamples at every step, so it does not apply to our sampler.

Output-wise subsampling is not privacy-free: if $\sum_t \mu_t^2$ is large, the influence of an included record accumulates before amplification. This motivates our hybrid denoiser, which switches to a public neural denoiser after the empirical denoising block so that the remaining generation is post-processing. We use the Edgeworth approximation only when the composed privacy loss is in a stable finite-sample regime; Appendix D states these diagnostics.

5 Related Work

Differentially private training has been the most direct approach to reduce memorization in diffusion models. DP training of diffusion models builds on DP-SGD (Abadi et al., 2016; Dockhorn et al., 2023; Ghalebikesabi et al., 2023; Liu et al., 2024) and protects the weights at a cost in sample quality. Private sampling has mostly been studied for Bayesian posterior inference, where the samples are model parameters rather than data (Welling & Teh, 2011; Wang et al., 2015; Li et al., 2019; Räisä et al., 2021; Zhang & Zhang, 2023). Bertazzi et al. (2025) show that privacy loss in SGLD can concentrate in the final step, but their analysis needs strongly convex, Lipschitz objectives, which our empirical denoiser does not satisfy.

Sampling-time privacy for diffusion models. Closest to our setting, Sehwag et al. (2023) also analyze privacy at generation rather than training time, clip denoiser outputs, treat each step as a Gaussian mechanism, and delegate late denoising to a public model. Our analysis differs in three respects. First, their per-step sensitivity C/k arises from a sample-and-aggregate ensemble of k non-private models trained on disjoint shards, so the sensitivity floor is set by the ensemble size and private training is still required; we obtain sensitivity $2C/|\mathcal{D}|$ by clipping individual contributions to a single empirical average, with no training at all. Second, they adopt a per-sample private-prediction threat model and name composition across multiple released samples as open; we give an explicit joint-release guarantee over N_{out} outputs under standard DP, using subsampled f-DP with block-level amplification. Third, we localize privacy loss to the critical window and place the handoff at the measured crossover rather than at a fixed schedule fraction, validating the choice with a membership-inference proxy. Wei et al. (2023) study inherent privacy properties of *discrete* denoising diffusion and likewise find that leakage concentrates at particular stages of generation. Their mechanism is the discrete transition kernel; ours is the continuous Gaussian reverse SDE, where the noise already injected by the sampler serves as the mechanism noise. Our results are therefore a continuous-state counterpart to their staged view, and our claims are scoped accordingly.

Empirical denoisers. Niedoba et al. (2025) characterize the empirical denoiser in the *non-private* setting and analyze how trained neural denoisers depart from it. We build on this object directly: our contribution is the clipped, privatized variant and its sensitivity analysis, and their non-private characterization is complementary evidence for the proxy argument in Section 3.3.

6 Experiments

We study (i) full-batch versus subsampled accounting, (ii) where privacy loss accumulates relative to the critical window, and (iii) the privacy-utility trade-off of the hybrid sampler.

Implementation We implement our modified reverse denoising process in PyTorch Paszke et al. (2019) and categorize our experiments into two regimes: (i) **full-batch**, where the entire dataset is used in each denoising step, and (ii) **output-wise subsampled SDEdit (Meng et al., 2021)**, where a minibatch of private examples is subsampled once per generated output and reused for an empirical denoising block. We apply the private empirical denoising steps until time t_0, then add Gaussian noise with scale $(\sigma_{t_1}^2 - \sigma_{t_0}^2)^{1/2}$ to move the sample back to an SDEdit restart level $t_1 > t_0$, and finally complete generation with a public neural denoiser from t_1 to 0. Only the empirical subsampled block is privacy-accounted. Re-noising to a higher noise level lets the public neural denoiser correct artifacts introduced by the empirical step while incurring no additional privacy loss by post-processing.

We use the EDM schedule (Karras et al., 2022) with $\sigma_{\min} = 0.002$, $\sigma_{\max} = 80$, $\rho = 7$, $N = 50$ steps for full-batch runs and $N = 39$ for the hybrid pipeline, and the public normalization $\beta = 1$ (Appendices E and J).

Privacy Analysis Given the noise schedule and sensitivity, we calculate the corresponding μ-GDP for each diffusion step (3). For the full-batch case, we derive the overall μ-GDP using the n-fold composition of μ_i-GDP mechanisms. For the output-wise subsampled SDEdit hybrid, we compute privacy using the certified PLD accounting library (Doroshenko et al., 2022) with grid spacing $h = 10^{-5}$ and `log_mass_truncation_bound` $= -50$, and the Edgeworth approximation

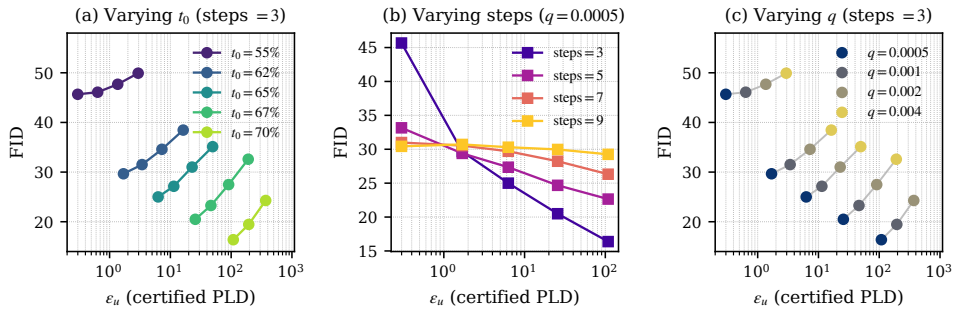


Figure 3: Privacy-utility trade-offs of the output-wise subsampled hybrid sampler on CIFAR-10. We report FID against the certified PLD upper bound ε_u ($\delta = 10^{-5}$, values from Table 8) while varying **(a)** the handoff time t_0, **(b)** the number of public re-noising steps from t_0 to t_1, and **(c)** the subsampling rate $q = B/n$.

Table 2: Comparison of DP generative models on CIFAR-10 and CelebA-64. Our method performs a short private editing block using an output-wise subsampled empirical denoiser, then hands off to a publicly pretrained neural denoiser (ImageNet for CIFAR-10, FFHQ for CelebA-64) to complete sampling. FID is computed for $N_{out} = 50,000$ samples. Lower FID is better.

CIFAR-10			**CelebA-64**		
Method	ε	FID	Method	ε	FID
DP-Diffusion (Ghalebikesabi et al., 2023)	10	9.8	DP-LDM (Liu et al., 2024)	10	14.3
DP-LDM (Liu et al., 2024)	10	8.4	DP-MEPF (Harder et al., 2023)	10	17.4
DP-API (Lin et al., 2023)	0.67	7.87	DP-DM (Dockhorn et al., 2023)	10	78.3
DP-MEPF (Harder et al., 2023)	10	29.1	DP-Promise (Wang et al., 2024)	10	25.3
DP-DM (Dockhorn et al., 2023)	10	97.7			
DP-Promise (Wang et al., 2024)	10	17.9			
Ours	PLD [11.14, 11.38] $\varepsilon_{ew} = 10$	27.1 ± 0.15	**Ours**	PLD [6.53, 6.77] $\varepsilon_{ew} = 5$	25.67 ± 0.03

from Section 4, which approximates the per-output privacy-loss distribution under both neighbouring-data hypotheses using its first four cumulants. The subsequent re-noising and public neural denoising stages incur no additional privacy loss.

Datasets and Evaluation On CIFAR-10 (Krizhevsky et al., 2009) ($n = 50{,}000$) the public denoiser is trained on ImageNet 32×32 (Russakovsky et al., 2015), and on CelebA 64×64 (Liu et al., 2015) ($n = 162{,}770$) it is trained on FFHQ (Karras et al., 2019). We use $\delta = 10^{-5} \approx 1/n$ for CIFAR-10 and 10^{-6} for CelebA-64 to match prior DP generative models, and report the stricter $\delta = 1/n^2$ in Appendix K. Utility is FID against the training set with $50{,}000$ samples, reported as mean $\pm$ standard deviation over three generation seeds where runs are repeated.

6.1 Privacy analysis of diffusion-based sampling with full training data

When every record is used at every step, a single sample already costs $\varepsilon \approx 200$ when denoising down to $\sigma_{\min} = 0.1$, which is needed for a recognizable sample, and at $\varepsilon \approx 1$ ($\sigma_{\min} = 5$) the sample is indistinguishable from noise (Figure 4, Appendix I). Since each denoising step operates on the full dataset, the per-sample privacy loss is already high, and composing multiple samples results in a rapidly growing total privacy budget under GDP. Subsampling can alleviate this issue through privacy amplification, making it more suitable for large-scale sample generation.

6.2 Hybrid denoising with output-wise subsampling

Figure 3 highlights three patterns. First, the empirical denoising stop timestep t_0 controls a clear privacy-utility tradeoff: stopping later at $t_0 = 70\%$ yields the best FID, while stopping earlier at $t_0 = 55\%$ yields the strongest privacy guarantee. Second, the best number of public re-noising steps depends on the privacy budget. With larger private blocks (65–70%, certified $\varepsilon \geq 6$) three re-noising steps give the best FID, while with the smallest block (55%, $\varepsilon \leq 0.3$) more steps help (FID 45.7 with

three steps and 30.4 with nine, $B = 25$). Re-noising steps are public, so they change utility but not ε. Third, decreasing the subsampling rate q lowers ε, as expected from privacy amplification. In our setting it also improves FID, likely because smaller q weakens the high-variance empirical denoiser before the public neural denoiser recovers global structure after re-noising.

To contextualize the effect of our privacy mechanisms, Table 2 compares the FID (Heusel et al., 2017) of our hybrid sampler with DP generative models on CIFAR-10 and CelebA-64, and Appendix J reports all hybrid configurations.

6.3 Privacy in relation to the critical windows

We extend our privacy analyses to quantify privacy levels based on where we stop relative to the critical window of the denoising process. Following Georgiev et al. (2023), we locate the critical window with a zero-shot CLIP classifier (Radford et al., 2021) applied along the trajectory (Figure 6(a)). The probability that the sample is classified as a ship is about 0.1 at $t = 0.27$, 0.3 at $t = 0.07$, and close to 1 at $t = 0.02$ (70%, 80%, and 88% of the $N = 50$ steps). Stopping the private block earlier gives a smaller μ but a noisier sample, and stopping later gives the reverse (Figure 7).

We choose the handoff point so that, over the private block, the empirical denoiser remains a faithful stand-in for the neural denoiser. The empirical denoiser is a window-local semantic proxy for the neural denoiser through the mid-noise window, which ends at $t = 1.98$, about half of the $N = 39$ steps (Table 1), and directional agreement stays at $\cos \geq 0.98$ through about 55% of generation (Figure 1, Table 4). A 55% handoff therefore keeps the private block essentially inside the proxy regime. Our 65% handoff ($t_0 \approx 0.52$) still comes before the CLIP-visible critical window (≈ 70–88%), but its private block covers most of the low-noise window, where substitution moves the endpoint more than re-drawing that window's noise ($1.64\times$ CLIP, Table 1). The proxy interpretation is therefore weaker for the 65% block, while the certified privacy guarantee holds for both, because it is a statement about the sampler we run. Stopping earlier tightens the certified bound at a utility cost, and stopping later extends the private block into the low-noise steps that dominate the privacy cost (Appendix J).

6.4 Privacy implications of neural and empirical denoisers in membership inference attacks

We compare the membership-inference success rate (ASR) implied by our ε for the subsampled empirical sampler (Appendix L) with the ASR that the attack of Duan et al. (2023) achieves on a neural denoiser trained on all of CIFAR-10, stopping at the same t and using the CIFAR-10 test set as non-members (Table 13, Appendix L). The attack is weak at every t (ASR at most 0.510) and stays below the bound for $q = 0.5$ at every t. At $q \leq 0.01$ it slightly exceeds the bound at $t = T/2$ (0.507 vs. at most 0.503), which is expected because the neural denoiser uses every record while the subsampled sampler includes each record with probability q.

7 Conclusion and Discussion

We give certified sampling-time privacy guarantees for continuous diffusion models by privatizing a clipped empirical denoiser. Framing each denoising step as a Gaussian mechanism, we account with GDP and f-DP, report certified (ϵ, δ) bounds from a PLD accountant, and characterize where Edgeworth approximation is and is not reliable. Our results show that leakage is highly non-uniform across timesteps, concentrating in critical windows where semantic structure emerges, while subsampling significantly amplifies privacy guarantees. These findings suggest that privacy in diffusion models is not solely determined by training but can be shaped by sampling strategies such as early stopping and hybrid use of public denoisers.

Our certificate covers the sampler we run. Samplers that query a neural denoiser trained on the private data at every step, and discrete diffusion, where the transition kernel would be the mechanism, remain open. Our sampler could also serve as a DP data-augmentation mechanism $\mathcal{A}(\mathcal{D})$: a model trained only on $\mathcal{A}(\mathcal{D})$ and public data is (ε, δ)-DP by post-processing, with no DP training algorithm. Since privacy loss concentrates at low noise, training on $\mathcal{A}(\mathcal{D})$ for $t \geq t_0$ and on public data for $t < t_0$ would spend the budget only on the formative regime.

AI USE STATEMENT

In this work, we used generative AI tools as writing, reviewing, and coding assistants. Specifically, we used them to improve grammar and word choice, to draft and revise parts of the text, to get feedback on drafts, such as consistency checks of numbers, claims, and references and suggestions for additional diagnostics. We also used them to search for and summarize related work, which we verified against the original sources, and to help write and debug code for the accounting, evaluation, and scripts, which we reviewed and tested before use. Additionally, we used AI to proofread our math equations and review our mathematical proofs. We did not use generative AI tools to generate research ideas or synthetic data, or to produce any reported number. Dataset cleaning, translation, and qualitative data analysis are not applicable to this work. We have reviewed all AI-assisted work and take full responsibility for the final content of this paper, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies privacy leakage from diffusion sampling and uses only public benchmark datasets (CIFAR-10, CelebA, ImageNet, FFHQ). Our guarantees cover the sampler in Algorithm 1 and should not be read as guarantees for other diffusion samplers (Appendix M).

REPRODUCIBILITY STATEMENT

Code is available at the anonymous link in Appendix A, including the PLD accounting, Edgeworth, trajectory-substitution, and FID scripts behind every table and figure. Algorithm 1 specifies the sampler, Section 3 and Appendix E give the sensitivity and the public normalization, and Appendix C gives full proofs. Section 6 and Appendix G list the accountant settings (sampling model, adjacency, q, N_{out}, δ, grid spacing, truncation, and precision). Appendix J lists every hyperparameter and reports all 95 CIFAR-10 and 112 CelebA-64 configurations (Tables 8 and 9) and the repeated-seed runs.

REFERENCES

Martin Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pp. 308–318, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341394. doi: 10.1145/2976749.2978318.

Andrea Bertazzi, Tim Johnston, Gareth O. Roberts, and Alain Oliviero Durmus. Differential privacy guarantees of markov chain monte carlo algorithms. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=lLnAua5poB.

Giulio Biroli, Tony Bonnaire, Valentin De Bortoli, and Marc Mézard. dynamical regimes of diffusion models. *arxiv preprint arxiv:2402.18491*, 2024.

Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *Proceedings of the 32nd USENIX Conference on Security Symposium*, SEC '23, USA, 2023. USENIX Association. ISBN 978-1-939133-37-3.

Tim Dockhorn, Tianshi Cao, Arash Vahdat, and Karsten Kreis. Differentially private diffusion models. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=ZPpQk7FJXF.

Jinshuo Dong, Aaron Roth, and Weijie J. Su. Gaussian differential privacy. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(1):3–37, 02 2022. ISSN 1369-7412. doi: 10.1111/rssb.12454. URL https://doi.org/10.1111/rssb.12454.

Vadym Doroshenko, Badih Ghazi, Pritish Kamath, Ravi Kumar, and Pasin Manurangsi. Connect the dots: Tighter discrete approximations of privacy loss distributions. *arXiv preprint arXiv:2207.04380*, 2022.

Jinhao Duan, Fei Kong, Shiqi Wang, Xiaoshuang Shi, and Kaidi Xu. Are diffusion models vulnerable to membership inference attacks? In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 8717–8730. PMLR, Jul 2023. URL https://proceedings.mlr.press/v202/duan23b.html.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference (TCC)*, pp. 265–284, 2006.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*, volume 57 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, New York, 1993.

Kristian Georgiev, Joshua Vendrow, Hadi Salman, Sung Min Park, and Aleksander Madry. The journey, not the destination: How data guides diffusion models. In *Arxiv preprint arXiv:2312.06205*, 2023.

Sahra Ghalebikesabi, Leonard Berrada, Sven Gowal, Ira Ktena, Robert Stanforth, Jamie Hayes, Soham De, Samuel L. Smith, Olivia Wiles, and Borja Balle. Differentially private diffusion models generate useful synthetic images, 2023.

Sivakanth Gopi, Yin Tat Lee, and Lukas Wutschitz. Numerical composition of differential privacy. *Advances in Neural Information Processing Systems*, 34:11631–11642, 2021.

Frederik Harder, Milad Jalali, Danica J. Sutherland, and Mijung Park. Pre-trained perceptual features improve differentially private image generation. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=R6W7zkMz0P.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Hailong Hu and Jun Pang. Membership inference of diffusion models, 2023.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4401–4410, 2019.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *advances in neural information processing systems*, volume 35, pp. 26565–26577, 2022.

Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011. doi: 10.1137/090756090. URL https://doi.org/10.1137/090756090.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, ON, Canada, 2009.

Bai Li, Changyou Chen, Hao Liu, and Lawrence Carin. On connecting stochastic gradient mcmc and differential privacy. In Kamalika Chaudhuri and Masashi Sugiyama (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 557–566. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/li19a.html.

Marvin Li and Sitan Chen. Critical windows: non-asymptotic theory for feature emergence in diffusion models. *arXiv preprint arXiv:2403.01633*, 2024.

Zinan Lin, Sivakanth Gopi, Janardhan Kulkarni, Harsha Nori, and Sergey Yekhanin. Differentially private synthetic data via foundation model apis 1: Images, 2023.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Michael F Liu, Saiyue Lyu, Margarita Vinaroz, and Mijung Park. Differentially private latent diffusion models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=AkdQ266kHj.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Andrew Lowy, Zhuohang Li, Jing Liu, Toshiaki Koike-Akino, Kieran Parsons, and Ye Wang. Why does differential privacy with large ε defend against practical membership inference attacks? In *AAAI Workshop on Privacy-Preserving Artificial Intelligence*, Feb 2024. URL https://www.merl.com/publications/TR2024-009. MERL Technical Report TR2024-009 / arXiv preprint arXiv:2402.09540.

Tomoya Matsumoto, Takayuki Miura, and Naoto Yanai. Membership inference attacks against diffusion models. In *2023 IEEE Security and Privacy Workshops (SPW)*, pp. 77–83, 2023. doi: 10.1109/SPW59333.2023.00013.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

Matthew Niedoba, Berend Zwartsenberg, Kevin Murphy, and Frank Wood. Towards a mechanistic explanation of diffusion model generalization. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *PMLR*, pp. 46389–46411, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Ossi Räisä, Antti Koskela, and Antti Honkela. Differentially private hamiltonian monte carlo. In *NeurIPS 2021 Workshop Privacy in Machine Learning*, 2021. URL https://openreview.net/forum?id=oqNqsnOVwlK.

Gabriel Raya and Luca Ambrogioni. Spontaneous symmetry breaking in generative diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

Antonio Sclocchi, Alessandro Favero, and Matthieu Wyart. a phase transition in diffusion models reveals the hierarchical nature of data. *arxiv preprint arxiv:2402.16991*, 2024.

Vikash Sehwag, Ashwinee Panda, Ashwini Pokle, Xinyu Tang, Saeed Mahloujifar, Mung Chiang, J Zico Kolter, and Prateek Mittal. Differentially private generation of high fidelity samples from diffusion models. In *ICML 2023 Workshop on Deployable Generative AI*, 2023. URL https://openreview.net/forum?id=vuVGcl0ed1.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. deep unsupervised learning using nonequilibrium thermodynamics. In *proceedings of the 32nd international conference on machine learning*, pp. 2256–2265, 2015.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Shuai Tang, Zhiwei Steven Wu, Sergul Aydore, Michael Kearns, and Aaron Roth. Membership inference attacks on diffusion models via quantile regression, 2024.

Aleksei Triastcyn and Boi Faltings. Bayesian differential privacy for machine learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9583–9592. PMLR, Jul 2020. URL https://proceedings.mlr.press/v119/triastcyn20a.html.

Haichen Wang, Shuchao Pang, Zhigang Lu, Yihang Rao, Yongbin Zhou, and Minhui Xue. dp-promise: Differentially private diffusion probabilistic models for image synthesis. In *33rd USENIX Security Symposium (USENIX Security 24)*, Philadelphia, PA, August 2024. USENIX Association. URL https://www.usenix.org/conference/usenixsecurity24/presentation/wang-haichen.

Yu-Xiang Wang, Stephen Fienberg, and Alex Smola. Privacy for free: Posterior sampling and stochastic gradient monte carlo. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2493–2502, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/wangg15.html.

Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled rényi differential privacy and analytical moments accountant. In *AISTATS*, 2019.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

Rongzhe Wei, Eleonora Kreačić, Haoyu Wang, Haoteng Yin, Eli Chien, Vamsi K Potluru, and Pan Li. On the inherent privacy properties of discrete denoising diffusion models. *arXiv preprint arXiv:2310.15524*, 2023.

Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pp. 681–688, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

Yixin Wu, Ning Yu, Zheng Li, Michael Backes, and Yang Zhang. Membership inference attacks against text-to-image generation models, 2023. URL https://openreview.net/forum?id=J41IW8Z7mE.

Wanrong Zhang and Ruqi Zhang. DP-fast MH: Private, fast, and accurate Metropolis-Hastings for large-scale Bayesian inference. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 41847–41860. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/zhang23aw.html.

Qinqing Zheng, Jinshuo Dong, Qi Long, and Weijie Su. Sharp composition bounds for gaussian differential privacy via edgeworth expansion. In *International Conference on Machine Learning*, pp. 11420–11435. PMLR, 2020.

A Code Availability

Code is available at: https://anonymous.4open.science/r/dp-denoiser-9612/

B Lipschitz-based sensitivity analysis for neural denoisers

One might wonder whether considering Lipschitz-based sensitivity analysis for neural denoisers would help validate the empirical denoiser bounds. We describe our thoughts on this below.

Let $\mathcal{L} : X^n \to \Theta$ be the score network learning algorithm which maps a dataset $S \in X^n$ to parameters $\theta \in \Theta$. Let's define the sensitivity of the score network as $\Delta_s = \sup_{x \in X} \left(\sup_{S \in X^n, i \in [n]} \|s_{\mathcal{L}(S)}(x) - s_{\mathcal{L}(S\setminus\{i\})}(x)\|_2 \right)$. If we could show that Δ_s is bounded, then the update step in sampling can be made differentially private with appropriate noise scales as follows: $\tilde{x}_t \leftarrow \tilde{x}_{t-1} + \frac{\sigma_i}{2} s_\theta(\tilde{x}_{t-1}, \sigma_i) + \mathcal{N}(0, \sigma_t^2 \Delta_s^2 I)$. The question is how to bound the sensitivity. Does the Lipschitz network help here?

Suppose the score net is a 1-Lipschitz network, satisfying $\|s_{\mathcal{L}(S)}(x) - s_{\mathcal{L}(S)}(x')\|_2 \leq \|x - x'\|_2$ and $\|s_{\mathcal{L}(S\setminus\{i\})}(x) - s_{\mathcal{L}(S\setminus\{i\})}(x')\|_2 \leq \|x - x'\|_2$. Using this we can bound the sensitivity, $\|s_{\mathcal{L}(\mathcal{S})}(x) - s_{\mathcal{L}(S\setminus\{i\})}(x)\|_2 \leq \|s_{\mathcal{L}(\mathcal{S})}(x) - s_{\mathcal{L}(\mathcal{S})}(x')\|_2 + \|s_{\mathcal{L}(\mathcal{S})}(x') - s_{\mathcal{L}(S\setminus\{i\})}(x)\|_2 \leq \|x - x'\|_2 + \|s_{\mathcal{L}(\mathcal{S})}(x') - s_{\mathcal{L}(S\setminus\{i\})}(x)\|_2$. So here even if $\|s_{\mathcal{L}(\mathcal{S})}(x') - s_{\mathcal{L}(S\setminus\{i\})}(x)\|_2$ is very small, we have $\max_{x,x'} \|x - x'\|_2$ on our upperbound.

Without having any knowledge other than all x, x' are pre-processed such that any of them is norm-bounded by L, the upperbound to this is $\max_{x,x'} \|x - x'\|_2 \leq \max_x \|x\|_2 \leq 2L$. In this case, the amount of noise we have to add in each denoising step is on the order of the entire input domain, and such an amount of noise will destroy the signal from the score net completely. Hence, this approach is not usable in practice.

C PROOFS FOR SUBSAMPLING PROPOSITIONS

We prove Proposition 4.1 and Proposition C.1 (per-step subsampling, stated below) under the sampling model stated in Section 4. The proofs are directional first, comparing two neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$ that differ in one record, and then symmetrized by taking both neighboring-data directions.

C.1 PROOF OF PROPOSITION 4.1

Fix one generated output and an empirical denoising block $\mathcal{B}$. Conditional on the differing record being included in the sampled rows, the privacy loss of the full block is the composition of Gaussian mechanisms with parameters $\{\mu_t : t \in \mathcal{B}\}$. By GDP composition, this conditional block is $\bar{\mu}_{\mathcal{B}}$-GDP, where

$$\bar{\mu}_{\mathcal{B}} = \left(\sum_{t \in \mathcal{B}} \mu_t^2 \right)^{1/2}. \tag{9}$$

The canonical binary experiment for this included-record event is therefore $P_{\mathcal{B}} = \mathcal{N}(0,1)$ versus $\mathcal{N}(\bar{\mu}_{\mathcal{B}}, 1)$. Under this model the differing record is not selected with probability $1 - q_{\mathcal{B}}$, in which case the two neighboring datasets induce the same output distribution and the trade-off curve is Id. It is selected with probability $q_{\mathcal{B}}$, in which case the trade-off curve is $G_{\bar{\mu}_{\mathcal{B}}}$. Thus the one-output directional trade-off curve is

$$h_{\mathcal{B}} = q_{\mathcal{B}} G_{\bar{\mu}_{\mathcal{B}}} + (1 - q_{\mathcal{B}}) \,\mathrm{Id}\,. \tag{10}$$

The N_{out} generated outputs use independent row-subsampling events, so f-DP composition gives the directional composed trade-off

$$g_{\mathrm{out}} = \bigotimes_{j=1}^{N_{\mathrm{out}}} h_{\mathcal{B}}. \tag{11}$$

Since differential privacy must hold for both neighboring-data directions, we symmetrize the directional curve and take the lower convex envelope, yielding

$$f_{\mathrm{out}} = \min\{g_{\mathrm{out}}, g_{\mathrm{out}}^{-1}\}^{**}. \tag{12}$$

Therefore the joint release is f_{out}-DP. Any public neural denoising or deterministic post-processing after the private block is post-processing of the same private release and cannot weaken the guarantee.

It remains to derive the (ε, δ) profile. In the above directional experiment, the privacy-loss random variable for one output can be written as

$$L_{\mathcal{B}}(z) = \log\left(1 - q_{\mathcal{B}} + q_{\mathcal{B}} \exp\left(\bar{\mu}_{\mathcal{B}} z - \frac{\bar{\mu}_{\mathcal{B}}^2}{2}\right)\right), \tag{13}$$

and the composed privacy loss is $S_{\mathrm{out}} = \sum_{j=1}^{N_{\mathrm{out}}} L_{\mathcal{B}}(Z_j)$. The hockey-stick divergence characterization of approximate DP gives

$$\delta_{\mathrm{out}}^{P \to Q}(\varepsilon) = \mathrm{Pr}_Q(S_{\mathrm{out}} > \varepsilon) - e^{\varepsilon} \mathrm{Pr}_P(S_{\mathrm{out}} > \varepsilon). \tag{14}$$

Applying the same argument in the reverse neighboring direction and taking the positive part gives the stated $\delta_{\mathrm{out}}(\varepsilon)$.

C.2 PER-STEP SUBSAMPLING

Proposition C.1 (Per-step subsampling)**.** *Consider instead a sampler that draws an independent minibatch at every private denoising step. For step t, let q_t be the inclusion probability and μ_t the Gaussian parameter conditional on inclusion, and set $h_t = q_t G_{\mu_t} + (1 - q_t)\,\mathrm{Id}$. For N_{out} outputs and T private steps, define*

$$g_{\mathrm{step}} = \bigotimes_{j=1}^{N_{\mathrm{out}}} \bigotimes_{t=1}^{T} h_t, \qquad f_{\mathrm{step}} = \min\{g_{\mathrm{step}}, g_{\mathrm{step}}^{-1}\}^{**}. \tag{15}$$

Then the joint release is f_{step}-DP. Hence, for every $\varepsilon \geq 0$ it is $(\varepsilon, \delta_{\mathrm{step}}(\varepsilon))$-DP, where $\delta_{\mathrm{step}}(\varepsilon) = \max_{\mathrm{dir}}[Pr_Q(S^{\mathrm{dir}}_{\mathrm{step}} > \varepsilon) - e^{\varepsilon} Pr_P(S^{\mathrm{dir}}_{\mathrm{step}} > \varepsilon)]_+$, $S_{\mathrm{step}} = \sum_{j=1}^{N_{\mathrm{out}}} \sum_{t=1}^{T} L_t(Z_{j,t})$, and $L_t(z) = \log(1 - q_t + q_t \exp(\mu_t z - \mu_t^2/2))$.

The two propositions describe different privacy-loss distributions. In general, $S_{\mathrm{step}} = \sum_{j,t} L_t(Z_{j,t}) \neq \sum_j L_{\mathcal{B}}(Z_j) = S_{\mathrm{out}}$, so $f_{\mathrm{step}} \neq f_{\mathrm{out}}$. Per-step amplification is valid only when the implementation truly resamples at every step. For our output-wise subsampled sampler, Proposition 4.1 is the correct accountant: amplification is applied once to the whole empirical block.

C.3 Proof of Proposition C.1

For per-step subsampling, the differing record is independently included at each private denoising step t. Conditional on inclusion at step t, the step is a Gaussian mechanism with GDP parameter μ_t, so the canonical included-record trade-off curve is G_{μ_t}. If the record is not included, the two neighboring datasets induce the same distribution for that step and the trade-off curve is Id. Therefore the one-step directional trade-off curve is

$$h_t = q_t G_{\mu_t} + (1 - q_t)\,\mathrm{Id}\,. \tag{16}$$

Because minibatches are redrawn independently across timesteps and generated outputs, the directional trade-off curve for N_{out} outputs and T private steps is the tensor-product composition

$$g_{\mathrm{step}} = \bigotimes_{j=1}^{N_{\mathrm{out}}} \bigotimes_{t=1}^{T} h_t. \tag{17}$$

Symmetrizing over the two neighboring-data directions and convexifying gives

$$f_{\mathrm{step}} = \min\{g_{\mathrm{step}}, g_{\mathrm{step}}^{-1}\}^{**}, \tag{18}$$

so the joint release is f_{step}-DP.

The corresponding likelihood-ratio representation is obtained by summing the independent per-step privacy losses. For one step,

$$L_t(z) = \log\left(1 - q_t + q_t \exp\left(\mu_t z - \frac{\mu_t^2}{2}\right)\right), \tag{19}$$

and hence

$$S_{\mathrm{step}} = \sum_{j=1}^{N_{\mathrm{out}}} \sum_{t=1}^{T} L_t(Z_{j,t}). \tag{20}$$

Using the same hockey-stick divergence conversion as above gives

$$\delta_{\mathrm{step}}(\varepsilon) = \max_{\mathrm{dir}} \left[\Pr_Q(S^{\mathrm{dir}}_{\mathrm{step}} > \varepsilon) - e^{\varepsilon} \Pr_P(S^{\mathrm{dir}}_{\mathrm{step}} > \varepsilon)\right]_+, \tag{21}$$

which is the claimed (ε, δ) guarantee.

D Diagnostics for Edgeworth Approximation

The standard Edgeworth approximation for subsampled Gaussian mechanisms is often stated for the setting where a fresh subsample is drawn at every private step. Our output-wise subsampling mechanism uses the same mixture privacy-loss model, but after grouping timesteps by independent subsampling events. Thus each "step" in the diagnostic below should be read as a block $\mathcal{B}$ with sampling probability $q_{\mathcal{B}}$ and Gaussian parameter

$$\bar{\mu}_{\mathcal{B}} = \left(\sum_{t \in \mathcal{B}} \mu_t^2\right)^{1/2}. \tag{22}$$

For a block, recall the privacy-loss variable

$$L_{\mathcal{B}}(z) = \log\left(1 - q_{\mathcal{B}} + q_{\mathcal{B}} \exp\left(\bar{\mu}_{\mathcal{B}} z - \frac{\bar{\mu}_{\mathcal{B}}^2}{2}\right)\right), \tag{23}$$

evaluated under both $P_{\mathcal{B}} = \mathcal{N}(0,1)$ and $Q_{\mathcal{B}} = (1-q_{\mathcal{B}})\mathcal{N}(0,1) + q_{\mathcal{B}}\mathcal{N}(\bar{\mu}_{\mathcal{B}}, 1)$. Let $\kappa_{r,\mathcal{B}}^D$ denote the r-th cumulant of $L_{\mathcal{B}}$ under $D \in \{P, Q\}$. Across generated samples and independent blocks, cumulants add:

$$M_D = \sum_{\mathcal{B}} \kappa_{1,\mathcal{B}}^D, \qquad V_D = \sum_{\mathcal{B}} \kappa_{2,\mathcal{B}}^D, \qquad K_{3,D} = \sum_{\mathcal{B}} \kappa_{3,\mathcal{B}}^D, \qquad K_{4,D} = \sum_{\mathcal{B}} \kappa_{4,\mathcal{B}}^D. \tag{24}$$

Edgeworth approximation. Let S denote a composed privacy-loss variable with mean M, variance V, third cumulant K_3, and fourth cumulant K_4 (that is, M_D, V_D, $K_{3,D}$, $K_{4,D}$ under $D \in \{P, Q\}$). The degree-two Edgeworth approximation to the CDF of $(S - M)/\sqrt{V}$ is

$$F_{\mathrm{EW}}(z) = \Phi(z) - \varphi(z)\left[\frac{K_3}{6V^{3/2}}(z^2 - 1) + \frac{K_4}{24V^2}(z^3 - 3z) + \frac{K_3^2}{72V^3}(z^5 - 10z^3 + 15z)\right], \tag{25}$$

where Φ and φ are the standard normal CDF and density. We apply this approximation separately under the two neighboring-data hypotheses, since the two privacy-loss CDFs need not match.

What the cumulants represent. For a random variable X, the cumulant generating function is

$$K_X(s) = \log \mathbb{E}[e^{sX}], \tag{26}$$

and the r-th cumulant is the r-th derivative of K_X at $s = 0$. In our accountant, $X = L_{\mathcal{B}}$ under either P or Q. The first four cumulants have the familiar interpretations

$$\kappa_1 = \mathbb{E}[X], \tag{27}$$

$$\kappa_2 = \mathrm{Var}(X), \tag{28}$$

$$\kappa_3 = \mathbb{E}[(X - \mathbb{E}X)^3], \tag{29}$$

$$\kappa_4 = \mathbb{E}[(X - \mathbb{E}X)^4] - 3\mathrm{Var}(X)^2. \tag{30}$$

Thus κ_1 and κ_2 give the mean and variance of the privacy loss, while κ_3 and κ_4 quantify the leading skewness and kurtosis corrections to the Gaussian approximation.

No lower bound on the sampling rate alone. A small sampling probability $q_{\mathcal{B}}$ does not by itself invalidate the Edgeworth approximation. For fixed finite $\bar{\mu}_{\mathcal{B}}$, the variance and higher cumulants shrink with powers of $q_{\mathcal{B}}$, and these powers largely cancel after standardization. The relevant question is not whether $q_{\mathcal{B}}$ is large, but whether the composed privacy-loss sum has enough effective variance and is not dominated by a small number of blocks.

Variance and dominance checks. First, V_P and V_Q should be positive and numerically stable. If either variance is extremely small, the mechanism may be close to perfect privacy, but the standardized Edgeworth CDF and inverse-CDF calculations become ill-conditioned. Second, no block should dominate the variance:

$$\max_{\mathcal{B}} \frac{\kappa_{2,\mathcal{B}}^P}{V_P} \ll 1, \qquad \max_{\mathcal{B}} \frac{\kappa_{2,\mathcal{B}}^Q}{V_Q} \ll 1. \tag{31}$$

This condition is particularly important for diffusion schedules, where late low-noise timesteps can dominate $\sum_t \mu_t^2$ if they are grouped into a large private block.

Standardized cumulants. The Edgeworth correction should be small at the quantiles used for the reported (ε, δ) guarantee. We monitor the standardized third and fourth cumulants

$$\rho_{3,P} = \frac{|K_{3,P}|}{V_P^{3/2}}, \qquad \rho_{4,P} = \frac{|K_{4,P}|}{V_P^2}, \qquad \rho_{3,Q} = \frac{|K_{3,Q}|}{V_Q^{3/2}}, \qquad \rho_{4,Q} = \frac{|K_{4,Q}|}{V_Q^2}. \tag{32}$$

Small values near the center of the distribution are not sufficient by themselves: degree-two Edgeworth terms multiply these quantities by polynomials in the standardized quantile z, so far-tail evaluations can still be unreliable.

Algorithm 1 Output-wise subsampled empirical denoising sampler

Require: $\mathcal{D}$ with $n = |\mathcal{D}|$, subsample size B, clip norm C, EDM schedule with switch time t_0 and restart level $t_1 > t_0$, public denoiser D_{pub} (data disjoint from $\mathcal{D}$), outputs N_{out}.
Ensure: N_{out} samples with an f_{out}-DP guarantee on $\mathcal{D}$.
1: $q_{\mathcal{B}} \leftarrow B/n$; $\mathcal{B} \leftarrow \{\, t : t_0 < t \le t_{\max} \,\}$ ▷ block inclusion prob.; private block
2: **for** $j = 1$ **to** N_{out} **do**
3: draw $S_j \subseteq \mathcal{D}$, $|S_j| = B$, and **reuse for all of** $\mathcal{B}$; $x \leftarrow \sigma_{\max}\epsilon$ ▷ subsample *once* per output
4: **for** $t \in \mathcal{B}$ in decreasing order **do**
5: $\hat{\mathbb{E}} \leftarrow \frac{1}{\beta|S_j|}\sum_{x_0 \in S_j} \bar{v}(x_0; x)$, $\bar{v} = v/\max(1, \|v\|/C)$, $v = x_0\, p_t(x \mid x_0)$ ▷ clipped direction
6: $x \leftarrow \big(1 - \frac{2\Delta t}{t}\big)x + \frac{2\Delta t}{t}\hat{\mathbb{E}} + \sqrt{2t\Delta t}\,\epsilon$ ▷ Gaussian mechanism, μ_t by eq. 6
7: **end for**
8: $x \leftarrow x + (\sigma_{t_1}^2 - \sigma_{t_0}^2)^{1/2}\epsilon$; $x^{(j)} \leftarrow D_{\text{pub}}$ integrated from t_1 to 0 ▷ re-noise; public refinement (post-processing)
9: **end for**
10: $\bar{\mu}_{\mathcal{B}} \leftarrow (\sum_{t\in\mathcal{B}} \mu_t^2)^{1/2}$; amplify *once* by $q_{\mathcal{B}}$ and compose over N_{out} outputs (Prop. 4.1) to obtain f_{out}.
11: **return** $\{x^{(j)}\}_{j=1}^{N_{\text{out}}}$ with certificate f_{out}

Tail and CDF checks. Very small δ values may require evaluating the approximation close to the boundary of the CDF, where finite-sample Edgeworth approximations are less stable. We therefore check that the approximated CDFs are monotone, remain in $[0, 1]$ before substantial clipping or projection, and produce a nonnegative $\delta(\varepsilon)$ curve in the reported range. We compute both neighboring-data directions and use the larger value. The Edgeworth value is never a certified bound. These diagnostics only decide whether the estimate is stable enough to show next to the certified PLD value, and configurations that fail them are marked invalid (Table 9).

Practical implications of output-wise subsampling. Suppose the dataset has size $n = |\mathcal{D}|$ and each generated sample draws a minibatch of size B under the sampling model of Section 4, reused for an empirical denoising block. For a fixed training record, the block-level inclusion probability is

$$q_{\mathcal{B}} = B/n. \tag{33}$$

For our CIFAR-10 setting with $n = 50{,}000$ and $B = 25$, this gives $q_{\mathcal{B}} = 5 \times 10^{-4}$. If we release $N_{\text{out}} = 50{,}000$ generated samples, then a fixed record is included in about

$$N_{\text{out}} q_{\mathcal{B}} = 25 \tag{34}$$

generated trajectories on average. The privacy-relevant point is that each inclusion can affect the entire empirical block. Therefore the output-wise subsampled accountant is not

$$N_{\text{out}} T \text{ subsampled Gaussian mechanisms with parameters } (q_t, \mu_t), \tag{35}$$

which would correspond to independently resampling at every denoising step. It is instead

$$N_{\text{out}} \text{ subsampled block mechanisms with parameters } \left(q_{\mathcal{B}}, \left(\sum_{t\in\mathcal{B}} \mu_t^2\right)^{1/2}\right). \tag{36}$$

This is the practical reason to subsample per output once and keep the empirical block short: output-wise subsampling sharply limits how often any record is accessed, but once a record is selected its influence accumulates across all empirical denoising steps in that block.

E Effect of the public normalization constant β

For EDM, the forward transition kernel is $p(x_t \mid x_0) = \mathcal{N}(x_t; x_0, t^2 I)$, so the marginal at each time is $\widehat{p}_t(x_t) = \frac{1}{|\mathcal{D}|}\sum_{x_0\in\mathcal{D}} p(x_t \mid x_0)$. The exact empirical posterior mean divides the weighted average by this data-dependent normalizer:

$$\widehat{E}[x_0 \mid x_t] = \frac{1}{\widehat{p}_t(x_t)\,|\mathcal{D}|}\sum_{x_0\in\mathcal{D}} x_0\, p(x_t \mid x_0). \tag{37}$$

Privately estimating and using this normalizer would require accounting for an additional data-dependent statistic. Our implemented mechanism instead fixes a public constant (hyperparameter) β and uses

$$\widehat{E}_\beta[x_0 \mid x_t] = \frac{1}{\beta\,|\mathcal{D}|} \sum_{x_0 \in \mathcal{D}} x_0\, p(x_t \mid x_0). \tag{38}$$

Consequently, $\beta = 1$ defines a deliberately *unnormalized empirical mechanism*; it is not an estimate that the density $\widehat{p}_t(x_t)$ numerically equals one. With public β, this remains a well-defined randomized mechanism with a rescaled empirical drift, and the privacy guarantee applies exactly to the implemented mechanism. Its privacy and utility both depend on the public scaling through

$$\mu_t(\beta) = \frac{2C}{\beta\,|\mathcal{D}|}\sqrt{\frac{2\Delta t}{t^3}}, \tag{39}$$

where larger β reduces both the private drift and μ_t, while smaller β strengthens both.

Table 3 reports β directly (the experimental code uses the inverse, $1/\beta$). The setup is CIFAR-10, the 65% empirical block, batch size 50, three-step SDEdit restart, $\delta = 10^{-5}$, and $N_{\text{out}} = 50{,}000$.

Table 3: Effect of the public normalization β on privacy and utility.

β	Edgeworth estimate	Certified PLD upper bound	FID
4	0.438	0.438	132.89
2	1.218	1.217	62.81
1	10.777	11.381	27.13
0.5	256.624	259.761	53.54
0.25	unstable	unstable	81.43

This ablation shows that the public normalization is a genuine privacy–utility parameter rather than a privacy-neutral approximation: a weak empirical drift gives strong privacy but poor utility, whereas an overly strong drift worsens the certified privacy value and destabilizes generation.

F Per-step magnitude gap between empirical and neural denoisers

The relevant diagnostic in the main text is Figure 1 (directional cosine similarity), which we supplement here with the mean-squared error (MSE) between the neural and empirical denoiser outputs along the trajectory.

Table 4: Directional cosine similarity (Fig. 1) and per-step MSE $\|\widehat{\mathbb{E}} - D\|^2$ between empirical and neural denoiser outputs, along the sampling trajectory. Rows are labelled by the denoiser active at that noise level under the hybrid sampler, and **switch** marks the empirical→public handoff for the $55/65/70\%$ configurations. Rows use the $N = 50$ EDM grid (percent = step$/N$). On the $N = 39$ grid of the hybrid runs these handoffs fall near $t = 1.5$, 0.52, and 0.28. MSE is mean $\pm$ standard error over 3 seeds.

t	% gen.	log SNR	$\cos(x_t - \widehat{\mathbb{E}},\, x_t - D)$	MSE	denoiser
80.0	0%	-10.2	1.0000	3.24 ± 0.03	emp.
10.2	32%	-6.0	1.0000	1.84 ± 0.07	emp.
3.29	46%	-3.8	0.9965	14.63 ± 0.20	emp.
1.29	55%	-1.9	0.9895	10.13 ± 0.09	**switch 55%**
0.55	65%	-0.2	0.9766	6.40 ± 0.05	**switch 65%**
0.27	70%	$+1.3$	0.9568	4.23 ± 0.04	**switch 70%**
0.09	78%	$+3.4$	0.9011	2.13 ± 0.02	pub.
0.02	88%	$+6.7$	0.7517	0.62 ± 0.004	pub.
0.002	98%	$+11.0$	0.3901	0.10 ± 0.0003	pub.

Cosine and MSE measure different quantities: MSE is $\|\widehat{\mathbb{E}} - D\|^2$, while Figure 1 reports the cosine between the residual directions $r_E = x_t - \widehat{\mathbb{E}}$ and $r_N = x_t - D$, related by $\|\widehat{\mathbb{E}} - D\|^2 =$

$\|r_E\|^2 + \|r_N\|^2 - 2\|r_E\|\|r_N\| \cos(r_E, r_N)$. The cosine can be near one while the magnitudes differ (large MSE), and at low noise both residual norms shrink, making the cosine numerically sensitive. Cosine alone therefore cannot establish closeness; we report MSE alongside it and narrow the proxy claim to directional agreement.

Absolute MSE is not comparable across rows: both residual norms shrink as $t \to 0$, so the small values at low noise reflect this scale collapse rather than closer agreement, and the only cross-row claim we make is the directional one.

The magnitude gap is largest in the mid-noise window (the 46–55% rows), yet Table 1 shows that the difference surviving to the released sample there stays below the noise-resampling reference: the private block sets the denoising *direction*, and the subsequent re-noising and public refinement (Sec. 6) repair magnitude at no additional privacy cost. Table 5 gives the absolute endpoint agreement behind Table 1, and Table 6 gives image-space agreement for both the substituted and the noise-resampled paths.

Table 5: Endpoint agreement between the all-neural reference and the path with the empirical denoiser substituted in one window (2048 paired trajectories, $N = 39$). CLIP and DINO are cosine similarities ($1-$ the swap distance of Table 1), with 95% bootstrap CIs in brackets. ΔCLIP is noise-control distance minus swap distance, with 95% paired-bootstrap CIs. Image-space metrics are in Table 6.

Window	t range	CLIP	DINO	ΔCLIP [95% CI]
high-noise	19.15–7.47	0.990 [0.989, 0.991]	0.928 [0.922, 0.934]	$+0.0821$ $[+0.0801, +0.0840]$
mid-noise	6.09–1.98	0.923 [0.921, 0.925]	0.483 [0.474, 0.491]	$+0.0228$ $[+0.0207, +0.0248]$
low-noise	1.54–0.38	0.887 [0.886, 0.889]	0.309 [0.303, 0.315]	-0.0441 $[-0.0459, -0.0422]$
min-noise	0.14–0.018	0.785 [0.783, 0.788]	0.124 [0.119, 0.129]	-0.1906 $[-0.1930, -0.1881]$
all steps	80–0.002	0.794 [0.792, 0.796]	0.121 [0.117, 0.126]	–

Table 6: Image-space agreement between the all-neural endpoint and each variant, from the same starting noise (2048 trajectories, $N = 39$, windows of 6 steps). *swap*: empirical denoiser substituted inside the window. *ref.*: neural denoiser kept and only that window's noise re-drawn. Higher means closer to the all-neural endpoint for all three metrics. Brackets are 95% paired bootstrap CIs.

Window (steps)		PSNR (dB)	SSIM	Label agreement
high-noise (9–14)	swap	33.19 [32.88, 33.50]	0.981 [0.979, 0.983]	0.867 [0.853, 0.882]
	ref.	11.77 [11.67, 11.86]	0.634 [0.629, 0.639]	0.412 [0.391, 0.433]
mid-noise (15–20)	swap	14.74 [14.63, 14.84]	0.731 [0.726, 0.736]	0.462 [0.440, 0.483]
	ref.	10.40 [10.36, 10.45]	0.466 [0.463, 0.469]	0.390 [0.368, 0.411]
low-noise (21–26)	swap	12.23 [12.19, 12.27]	0.412 [0.410, 0.415]	0.255 [0.236, 0.274]
	ref.	11.98 [11.94, 12.03]	0.415 [0.412, 0.419]	0.603 [0.582, 0.624]
min-noise (29–34)	swap	11.39 [11.36, 11.42]	0.115 [0.112, 0.118]	0.152 [0.136, 0.167]
	ref.	22.69 [22.67, 22.72]	0.917 [0.916, 0.918]	0.793 [0.776, 0.811]
all steps (0–38)	swap	10.19 [10.14, 10.23]	0.087 [0.085, 0.089]	0.171 [0.155, 0.188]

G Comparison between PLD Accountant and Edgeworth Approximation

G.1 Accounting Procedure

Shared steps. (1) Compute μ_t for every private step and $\bar{\mu}_{\mathcal{B}} = (\sum_{t \in \mathcal{B}} \mu_t^2)^{1/2}$ (Table 7 lists $\bar{\mu}_{\mathcal{B}}$ for each handoff). (2) Form the one-output pair $P_{\mathcal{B}} = \mathcal{N}(0, 1)$, $Q_{\mathcal{B}} = (1 - q_{\mathcal{B}})\mathcal{N}(0, 1) + q_{\mathcal{B}}\mathcal{N}(\bar{\mu}_{\mathcal{B}}, 1)$ and its privacy loss $L_{\mathcal{B}}$ (Proposition 4.1), in both directions ($Q_{\mathcal{B}}$ against $P_{\mathcal{B}}$ and $P_{\mathcal{B}}$ against $Q_{\mathcal{B}}$). For a pair (A, A') with privacy loss $\ell = \log(\mathrm{d}A/\mathrm{d}A')$, the hockey-stick divergence is $H_{e^\varepsilon}(A\|A') = \mathbb{E}_A[(1 - e^{\varepsilon - \ell})_+]$, so $\delta(\varepsilon)$ depends on the pair only through the distribution of ℓ under A, its PLD.

Table 7: Privacy accounting comparison for different empirical block sizes.

Empirical block (%)	$\bar{\mu}_{\mathcal{B}}$	Edgeworth estimate $\varepsilon_{\mathrm{EW}}$	PLD lower bound	PLD upper bound	RDP upper bound
55	0.69	0.2987	0.04880	0.2988	0.36
62	1.56	1.68	1.44	1.69	2.63
65	2.09	5.33	6.05	6.29	7.81
67	2.84	23.0	25.49	25.7	48.8

PLD accountant (certified, reported as ε_u). (3) For each direction, discretize the PLD of one output pessimistically on a grid of width $h = 10^{-5}$, so that the hockey-stick divergence of the discrete PLD upper-bounds the true one at every ε. Tail mass below e^{-50} is truncated and added to δ. (4) Compose the N_{out} independent outputs by convolving the discretized PLD with itself N_{out} times, computed with the FFT. Any mass truncated during composition is also added to δ. The result is a pessimistic PLD of S_{out}. (5) For each direction evaluate $\delta(\varepsilon) = \sum_s \omega(s)(1 - e^{\varepsilon - s})_+$ plus the truncated mass, where ω is the composed discrete PLD, take the larger of the two directions, and find the smallest ε with $\delta_{\mathrm{out}}(\varepsilon) \le \delta$. This is ε_u. (6) Repeating (3)–(5) with every value rounded down gives ε_ℓ, which is not a bound and only shows how much of ε_u is discretization slack. We use the implementation of Doroshenko et al. (2022) (Google `dp_accounting`) to automatically calculate these values given $\bar{\mu}_{\mathcal{B}}$ and subsampling rate.

Edgeworth approximation (estimate). (3′) For $D \in \{P, Q\}$, compute the first four cumulants $\kappa^D_{r,\mathcal{B}}$ of $L_{\mathcal{B}}$ under $D_{\mathcal{B}}$ by numerical integration over z. (4′) Cumulants add over independent outputs, so S_{out} has $M_D = N_{\mathrm{out}}\kappa^D_{1,\mathcal{B}}$, $V_D = N_{\mathrm{out}}\kappa^D_{2,\mathcal{B}}$, $K_{3,D} = N_{\mathrm{out}}\kappa^D_{3,\mathcal{B}}$, and $K_{4,D} = N_{\mathrm{out}}\kappa^D_{4,\mathcal{B}}$ (Appendix D). (5′) Approximate $\Pr_D(S_{\mathrm{out}} > \varepsilon) \approx 1 - F_{\mathrm{EW}}\big((\varepsilon - M_D)/\sqrt{V_D}\big)$ with eq. 25, insert both probabilities into $\delta_{\mathrm{out}}(\varepsilon)$ of Proposition 4.1 for each direction, take the larger, and solve $\delta_{\mathrm{out}}(\varepsilon) = \delta$ by bisection. (6′) Check the diagnostics of Appendix D. The value is an estimate: the truncated series has an error of unknown sign and there is no discretization or truncation control. The CLT column of Table 11 instead replaces the composed trade-off curve by a single Gaussian curve G_μ, with μ from the summed first two moments of the per-output privacy loss (Dong et al., 2022).

G.2 Comparison

The degree-two Edgeworth approximation used in the paper is a **numerical approximation**, not a one-sided certified DP upper bound. Although Propositions 4.1 and C.1 exactly characterize the relevant block/per-step privacy-loss distributions, replacing their CDFs by a truncated Edgeworth expansion introduces an approximation error whose sign is not controlled in our finite heterogeneous setting.

To provide a formal guarantee, we additionally composed the mechanism actually implemented—within a private block, conditional per-step Gaussian privacy accumulates to $\bar{\mu}_{\mathcal{B}} = (\sum_{t\in\mathcal{B}} \mu_t^2)^{1/2}$, subsampling amplification is applied **once** to the block, and the result is composed over N_{out} released samples—using PLD (Doroshenko et al., 2022) and RDP (Wang et al., 2019) accountants. All accountants use the sampling model of Section 4 ($q = B/n$, replace-one adjacency), block parameter $\bar{\mu}_{\mathcal{B}}$, and the same output count. RDP is a conservative certified relaxation; the PLD method provides a rigorous upper bound and a lower bound that bracket the optimal privacy value after controlling discretization and truncation error. At $\delta = 10^{-5}$, $N_{\mathrm{out}} = 50{,}000$, $B = 25$ for CIFAR-10, PLD grid spacing $h = 10^{-5}$ and `log_mass_truncation_bound` $= -50$:

The comparison reveals two regimes in these evaluated configurations. For $\varepsilon \lesssim 2$, the Edgeworth estimate lies within 1% of the certified PLD upper bound (55% and 62% blocks). For $\varepsilon \gtrsim 5$, it is **anti-conservative**: at the 65% and 67% blocks, the Edgeworth estimate falls below even the optimistic PLD *lower* estimate (5.33 vs. $[6.05, 6.29]$ and 22.98 vs. $[25.49, 25.70]$), understating the certified value by about 10–15%.

H Differential Privacy and Gaussian Differential Privacy

H.1 Differential Privacy

The concept of differential privacy (DP) (Abadi et al., 2016; Dwork et al., 2006) provides a formal guarantee that a model behaves almost identically regardless of whether any particular data sample is included in the training set. In other words, DP bounds the extent to which a single sample can influence the model's behavior. More precisely, a mechanism $\mathcal{M}$, which may be a training algorithm such as Stochastic Gradient Descent (SGD) or, more generally, any randomized algorithm including diffusion sampling, is said to be ϵ-DP if the privacy loss is bounded by ϵ: $\log \frac{Pr[\mathcal{M}(\mathcal{D})=o]}{Pr[\mathcal{M}(\mathcal{D}')=o]} \leq \epsilon$, where $\mathcal{D}$ and $\mathcal{D}'$ differ in the record of a single individual; and $\mathcal{M}(\mathcal{D}) = o$ denotes that applying the mechanism $\mathcal{M}$ on a dataset $\mathcal{D}$ ends up converging to the output event o.

For instance, if $\mathcal{M}$ is a training optimizer such as SGD, then $\mathcal{M}(\mathcal{D})$ is the trained model and $Pr(\mathcal{M}(\mathcal{D}) = o)$ depicts the probability the model o being the trained one. Intuitively, if deleting a single data point in $\mathcal{D}$ to obtain $\mathcal{D}'$ does not substantially change the probability of obtaining the same output model o, then the training algorithm can be regarded as privacy-preserving[1]. Put more simply, if the probability the model converges to the same parameters is relatively equivalent regardless of whether a particular sample is present, then that sample cannot be said to meaningfully affect training. More generally, $\mathcal{M}$ is (ε, δ)-DP if for all neighboring $\mathcal{D}, \mathcal{D}'$ and every measurable set of outputs O,

$$\Pr[\mathcal{M}(\mathcal{D}) \in O] \leq e^{\varepsilon} \Pr[\mathcal{M}(\mathcal{D}') \in O] + \delta. \tag{40}$$

The privacy-loss condition $\Pr\big[\log \frac{\Pr[\mathcal{M}(\mathcal{D})=o]}{\Pr[\mathcal{M}(\mathcal{D}')=o]} \geq \varepsilon\big] \leq \delta$ implies eq. 40 but is not equivalent to it.

A natural building block for designing private (either training or sampling) algorithms is the *Gaussian mechanism*. This is particularly relevant for our setting, since each step in diffusion sampling injects Gaussian noise. Formally, given a function $\mathbf{h} : \mathcal{D} \mapsto \mathbb{R}^p$, the *Gaussian mechanism* releases

$$\mathcal{M}_{\text{Gauss}}(\mathcal{D}; \mathbf{h}) = \mathbf{h}(\mathcal{D}) + n, . \tag{41}$$

where $n \sim \mathcal{N}(0, \sigma^2 \Delta_{\mathbf{h}}^2 \mathbf{I})$. Here, $\Delta_{\mathbf{h}}$ denotes the *global sensitivity*—the maximum L_2 difference between outputs on two neighboring datasets—and σ controls the noise level as a function of (ϵ, δ). By adding noise calibrated to sensitivity, the Gaussian mechanism ensures that the presence or absence of any single sample only has a limited influence on the output. This mechanism underlies differentially private training methods such as DP-SGD (Abadi et al., 2016), and, as we will show, it also directly connects to the diffusion sampling process we analyze in this work.

As an interesting fact in differential privacy, it is widely known that subsampling amplifies privacy (Kasiviswanathan et al., 2011). For example, if we flip a coin to decide whether each entry of the training data is in or out to form a smaller dataset, we would expect the mechanism using the smaller dataset to be twice as private as the original mechanism using the full dataset. With a flipping probability of 50%, every individual benefits from perfect privacy if the individual is not included in the smaller dataset. To benefit from the privacy amplification effect, a subsampled Gaussian mechanism (applying the Gaussian mechanism on randomly subsampled data) is also widely used (Abadi et al., 2016; Wang et al., 2019). We will also study the effect of subsampling on the privacy loss incurring during the diffusion sampling process.

H.2 Gaussian Differential Privacy

While the Gaussian mechanism provides a natural foundation for analyzing privacy, the classical (ϵ, δ)-DP framework yields only loose bounds when these mechanisms are composed, often resulting in substantially looser guarantees than the actual privacy loss. To address this, we adopt the more refined f-DP framework (Dong et al., 2022), which characterizes privacy through hypothesis testing and provides an exact trade-off curve (function), rather than the looser worst-case bounds of (ϵ, δ)-DP.

A particularly important special case is *Gaussian Differential Privacy (GDP)*, which describes the privacy profile of the Gaussian mechanism exactly using a single parameter μ. For example, in

[1] SGD and Adam are not DP algorithms as the gradient update may be dominated by a huge outlier that significantly affects the model's parameters. As a result, a model trained with SGD/Adam can potentially reveal information about such outliers, leaving the data vulnerable to extraction attacks.

the case of eq. 41 with $n \sim \mathcal{N}(0, \sigma^2 \Delta_{\mathbf{h}}^2 \mathbf{I})$, the Gaussian mechanism corresponds to $\frac{1}{\sigma}$-GDP. For a Gaussian mechanism, we can define a trade-off function T using the standard normal CDF Φ given by,

$$G_\mu := T(\mathcal{N}(0,1), \mathcal{N}(\mu,1). \quad G_\mu(\alpha) := \Phi(\Phi^{-1}(1-\alpha) - \mu). \tag{42}$$

This makes GDP especially well aligned with diffusion sampling, where Gaussian perturbations are intrinsic to every denoising step. Moreover, GDP is closed under composition (Corollary 2 of Dong et al. (2022)): the cumulative privacy loss of n mechanisms with parameters $\mu_1, \ldots, \mu_n$ is given tightly by

$$\mu = \sqrt{\sum_{i=1}^{n} \mu_i^2}. \tag{43}$$

Thus, GDP not only matches the Gaussian mechanism exactly but also enables precise accounting of privacy loss across many Gaussian steps that constitute diffusion sampling. A μ-GDP mechanism is $(\varepsilon, \delta(\varepsilon))$-DP for every $\varepsilon \geq 0$ with the closed form (Dong et al., 2022)

$$\delta(\varepsilon) = \Phi\left(-\frac{\varepsilon}{\mu} + \frac{\mu}{2}\right) - e^{\varepsilon}\Phi\left(-\frac{\varepsilon}{\mu} - \frac{\mu}{2}\right), \tag{44}$$

which we use for all full-batch conversions.

For subsampled Gaussian mechanisms, the trade-off function in eq. 42 needs to be modified to incorporate the randomness coming from the subsampling process and the GDP composition in eq. 43 is no longer valid. In this case, one can benefit from using the central limit theorem, which tells us that composing many f-DP mechanisms converges to GDP, as formally stated in Theorem 3.5 of Dong et al. (2022).

I FULL-BATCH SAMPLING

In the full-batch regime every record contributes to every denoising step, so the per-step parameters μ_t compose exactly by eq. 3 and convert to (ε, δ) by eq. 44. We vary the minimum noise level $\sigma_{\min}$ to understand its impact on privacy guarantees. Figure 5 shows how the per-step μ grows as the noise level falls.

Lower $\sigma_{\min}$ corresponds to less privacy. Figure 4 shows how the choice of $\sigma_{\min}$ affects the privacy-utility trade-off. As $\sigma_{\min}$ decreases (getting closer to the training data), the GDP analysis gives worse (ε, δ)-DP guarantees.

The full-batch empirical denoiser does not give good utility for its privacy cost. At $\varepsilon \approx 1$ (corresponding to $\sigma_{\min} = 5$), the generated sample is hard to distinguish from pure noise (Figure 4, right). For a recognizable sample we need $\sigma_{\min} = 0.1$, which corresponds to $\varepsilon \approx 200$.

Scalability of full-batch sampling. Even at low $\sigma_{\min}$, full-batch diffusion sampling does not scale well with the number of generated samples. Since each denoising step operates on the full dataset, the per-sample privacy loss is already high, and composing multiple samples results in a rapidly growing total privacy budget under GDP. Subsampling alleviates this through privacy amplification (Section 4).

I.1 PRIVACY LOSS IN RELATION TO CRITICAL WINDOWS

Figure 6 shows the CLIP-based detection of the critical window used in Section 6.3, and Figure 7 shows (ϵ, δ)-DP traces (left) and Type I and II error profiles (right) under GDP with varying stopping time points in the reverse process.

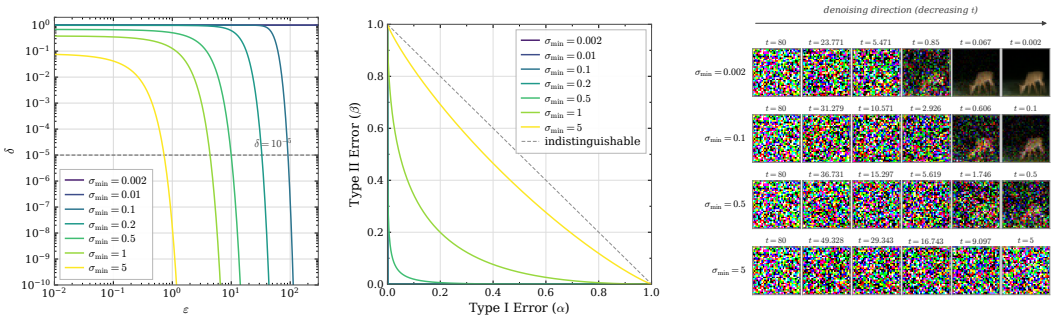


Figure 4: GDP Analysis of Diffusion-Based Sampling (full batch) and Corresponding Diffusion Process Visualization. The noise scheduling constants are $\sigma_{max} = 80$, $N = 50$, $\beta = 1$, and $\rho = 7$. The (ϵ, δ)-DP guarantees account for a single sample. **Left.** GDP conversion to (ϵ, δ)-DP for varying $\sigma_{\min}$. **Middle.** Type I and Type II Error under full-batch GDP with varying σ_{min}. **Right.** Visualization of Diffusion Process with Varying $\sigma_{\min}$. Only six timesteps are shown for figure clarity.

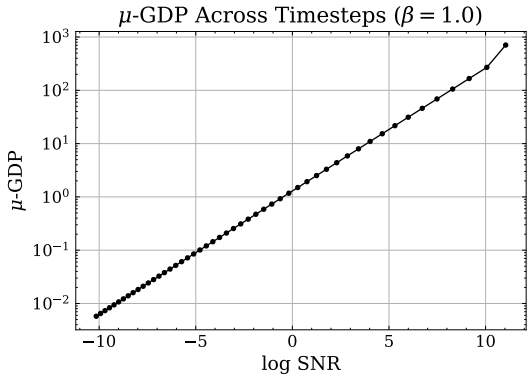


Figure 5: μ-GDP for each time step with a full batch of data.

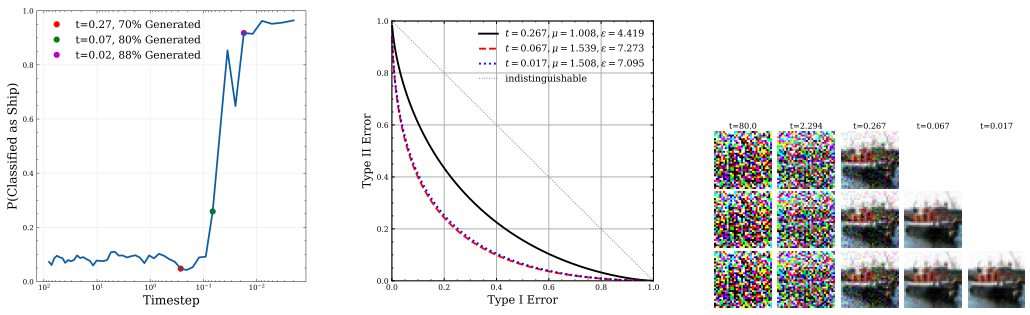
J Evaluation of our Hybrid Empirical Denoiser and Neural Denoiser on CIFAR-10 and CelebA

The CIFAR-10 experiment uses an ImageNet-pretrained diffusion model for the neural denoiser, and we adopt the ρ-power schedule of EDM (Karras et al., 2022). The scheduler is parameterized by the minimum and maximum noise levels $(\sigma_{min}, \sigma_{max})$, the noise decay rate ρ, and the number of denoising steps N. We experiment with $\sigma_{\min} = 0.002$, $\sigma_{\max} = 80$, $\rho = 7$, $N = 39$, and the public normalization $\beta = 1$ (Appendix E).

We conduct a hyperparameter search over the number of re-noising steps $T \in \{1, 3, 5, 7, 9\}$, empirical generation proportion $prop \in \{55, 62, 65, 67, 70\}\%$, and subsample batch size $B \in \{25, 50, 100, 200\}$, giving 95 configurations (the 70%, $B = 200$ setting is not reported due to overflowing privacy calculations, with results in Table 8. For SDEdit with output-wise subsampling of size B, we run empirical denoising for the first $prop\%$ of the generation process, up to t_0. Afterwards, we re-noise by adding Gaussian noise with scale $(\sigma_{t_1}^2 - \sigma_{t_0}^2)^{1/2}$. We continue denoising with the public Imagenet denoiser starting at t_1 until the end. We generate 50,000 samples for every run and compute the FID against CIFAR10.

Experiments were conducted on NVIDIA L40S GPUs with 48GB memory. Since our method primarily performs private sampling than full diffusion training, compute requirements were moderate. CIFAR-10 experiments typically required approximately 2.5 GPU-hours per run.

For CelebA (Liu et al., 2015) at 64×64 ($n = 162{,}770$, $\delta = 10^{-6}$), we use FFHQ (Karras et al., 2019), trained without access to the private CelebA training split, as the disjoint public source for the neural denoiser, and keep the same EDM scheduler ($\sigma_{min} = 0.002$, $\sigma_{max} = 80$, $\rho = 7$, $N = 39$)



(a) (b) (c)

Figure 6: Detection of critical window. **(a)** zero-shot CLIP model detects the probability of an image being generated belonging to the *ship* class. **(b)** Privacy costs at three time points. **(c)** Generated images denoised at varying timesteps.

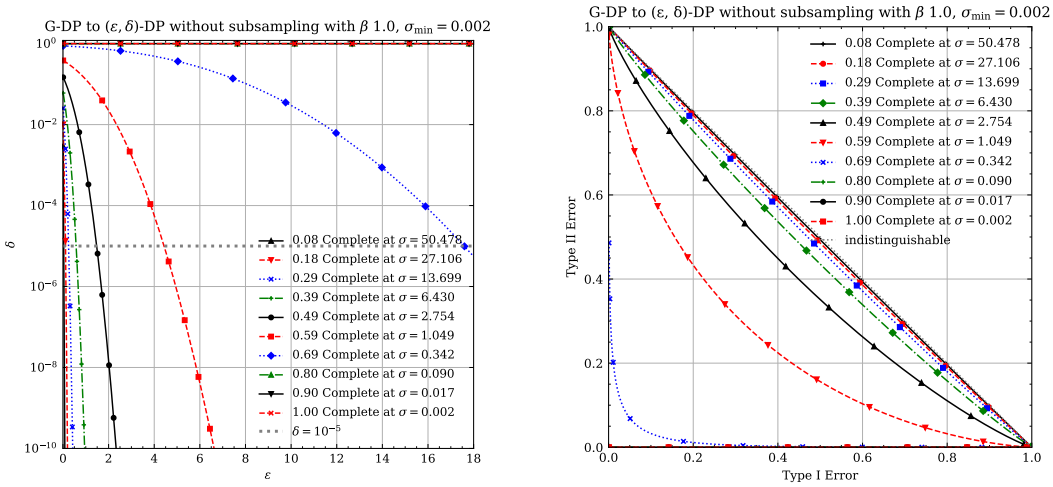


Figure 7: (ϵ, δ)-DP traces (Left) and Type I and II error profiles (Right) under GDP with varying stopping time points in the reverse process

and public normalization $\beta = 1$. We sweep the same base grid of re-noising steps $T \in \{1, 3, 5, 7, 9\}$, empirical generation proportion $prop \in \{55, 62, 65, 67, 70\}\%$, and subsample batch size $B \in \{25, 50, 100, 200\}$, and at $T = 3$ additionally refine B (to $\{60, 70\}$ for $prop \in \{62, 65\}\%$ and to $\{80, 90, 120, 140\}$ for $prop \in \{65, 67\}\%$), giving 112 configurations in total, with results reported in Table 9. The pipeline is otherwise identical to the CIFAR-10 setting: we run empirical denoising for the first $prop\%$ of generation up to t_0, re-noise by adding Gaussian noise with scale $(\sigma_{t_1}^2 - \sigma_{t_0}^2)^{1/2}$, and complete generation with the public FFHQ denoiser from t_1 until the end. We generate 50,000 samples for every run and compute the FID against CelebA. The Edgeworth diagnostic ε_{sym} is available only for the refined $T = 3$ rows and is marked invalid where the reuse-validity flag is false or a cumulant ratio $\rho \geq 1$ (e.g., $prop = 65\%$, $B = 60$, where $\rho_4^Q = 1.19$). Table 10 repeats the $B = 100$, $T = 3$ handoff sweep over three generation seeds.

Table 8: Hyperparameter search for output-wise subsample SDEdit on CIFAR-10. We sweep the number of re-noising steps $T \in \{1, 3, 5, 7, 9\}$, the proportion of empirical (denoiser) generation $p \in \{55, 62, 65, 67, 70\}\%$, and the subsample batch size $B \in \{25, 50, 100, 200\}$; 95 configurations in total. FID measures sample quality (lower is better). PLD $[\varepsilon_\ell, \varepsilon_u]$ is the PLD interval under the sampling model of Section 4 (optimistic lower / pessimistic upper estimate, Google `dp_accounting`). Only ε_u is a certified bound. $\varepsilon_{\mathrm{sym}}$ is the second degree Edgeworth approximation (lower is better).

Noise Steps	**Prop. Empirical (%)**	**Subsample Batch Size**	**FID** $\downarrow$	**PLD** $[\varepsilon_\ell, \varepsilon_u] \downarrow$	$\varepsilon_{\mathbf{sym}} \downarrow$
1	55	25	147.43	[0.049, 0.299]	0.299
	55	50	143.57	[0.386, 0.636]	0.636
	55	100	141.23	[1.111, 1.361]	1.360
	55	200	139.74	[2.702, 2.952]	2.946
	62	25	78.79	[1.442, 1.691]	1.683
	62	50	78.23	[3.172, 3.422]	3.513
	62	100	78.73	[7.009, 7.259]	7.515
	62	200	80.37	[16.009, 16.259]	16.706
	65	25	78.53	[6.052, 6.289]	5.331
	65	50	77.65	[11.136, 11.381]	10.777
	65	100	78.24	[22.530, 22.780]	22.540
	65	200	79.77	[49.332, 49.582]	49.625
	67	25	80.33	[25.486, 25.696]	22.982
	67	50	80.59	[46.196, 46.438]	44.481
	67	100	81.94	[90.634, 90.898]	89.733
	67	200	84.24	[190.673, 190.915]	190.198
	70	25	74.98	[108.154, 108.444]	104.035
	70	50	76.42	[194.318, 194.645]	191.700
	70	100	79.52	[369.696, 370.025]	368.188
3	55	25	45.67	[0.049, 0.299]	0.299
	55	50	46.09	[0.386, 0.636]	0.636
	55	100	47.69	[1.111, 1.361]	1.360
	55	200	49.92	[2.702, 2.952]	2.946
	62	25	29.66	[1.442, 1.691]	1.683
	62	50	31.51	[3.172, 3.422]	3.513
	62	100	34.59	[7.009, 7.259]	7.515
	62	200	38.46	[16.009, 16.259]	16.706
	65	25	25.00	[6.052, 6.289]	5.331
	65	50	27.13	[11.136, 11.381]	10.777
	65	100	31.01	[22.530, 22.780]	22.540
	65	200	35.12	[49.332, 49.582]	49.625
	67	25	20.49	[25.486, 25.696]	22.982
	67	50	23.27	[46.196, 46.438]	44.481
	67	100	27.48	[90.634, 90.898]	89.733
	67	200	32.59	[190.673, 190.915]	190.198
	70	25	16.38	[108.154, 108.444]	104.035
	70	50	19.46	[194.318, 194.645]	191.700
	70	100	24.28	[369.696, 370.025]	368.188
5	55	25	33.16	[0.049, 0.299]	0.299
	55	50	34.12	[0.386, 0.636]	0.636
	55	100	36.19	[1.111, 1.361]	1.360
	55	200	38.89	[2.702, 2.952]	2.946
	62	25	29.39	[1.442, 1.691]	1.683
	62	50	30.94	[3.172, 3.422]	3.513
	62	100	33.94	[7.009, 7.259]	7.515
	62	200	37.44	[16.009, 16.259]	16.706
	65	25	27.35	[6.052, 6.289]	5.331
	65	50	29.12	[11.136, 11.381]	10.777
	65	100	32.55	[22.530, 22.780]	22.540
	65	200	36.30	[49.332, 49.582]	49.625
	67	25	24.68	[25.486, 25.696]	22.982
	67	50	26.94	[46.196, 46.438]	44.481
	67	100	30.70	[90.634, 90.898]	89.733
	67	200	34.91	[190.673, 190.915]	190.198
	70	25	22.65	[108.154, 108.444]	104.035
	70	50	25.21	[194.318, 194.645]	191.700
	70	100	29.07	[369.696, 370.025]	368.188

Continued on next page

Table 8 continued

Noise Steps	Prop. Empirical (%)	Subsample Batch Size	FID ↓	PLD $[\varepsilon_\ell, \varepsilon_u]$ ↓	$\varepsilon_{\mathbf{sym}}$ ↓
7	55	25	30.99	[0.049, 0.299]	0.299
	55	50	31.97	[0.386, 0.636]	0.636
	55	100	34.01	[1.111, 1.361]	1.360
	55	200	36.38	[2.702, 2.952]	2.946
	62	25	30.54	[1.442, 1.691]	1.683
	62	50	31.81	[3.172, 3.422]	3.513
	62	100	34.34	[7.009, 7.259]	7.515
	62	200	37.34	[16.009, 16.259]	16.706
	65	25	29.70	[6.052, 6.289]	5.331
	65	50	31.26	[11.136, 11.381]	10.777
	65	100	34.02	[22.530, 22.780]	22.540
	65	200	37.18	[49.332, 49.582]	49.625
	67	25	28.23	[25.486, 25.696]	22.982
	67	50	29.86	[46.196, 46.438]	44.481
	67	100	33.13	[90.634, 90.898]	89.733
	67	200	36.41	[190.673, 190.915]	190.198
	70	25	26.32	[108.154, 108.444]	104.035
	70	50	28.51	[194.318, 194.645]	191.700
	70	100	31.76	[369.696, 370.025]	368.188
9	55	25	30.43	[0.049, 0.299]	0.299
	55	50	31.41	[0.386, 0.636]	0.636
	55	100	33.14	[1.111, 1.361]	1.360
	55	200	35.07	[2.702, 2.952]	2.946
	62	25	30.71	[1.442, 1.691]	1.683
	62	50	31.79	[3.172, 3.422]	3.513
	62	100	33.94	[7.009, 7.259]	7.515
	62	200	36.49	[16.009, 16.259]	16.706
	65	25	30.30	[6.052, 6.289]	5.331
	65	50	31.80	[11.136, 11.381]	10.777
	65	100	34.00	[22.530, 22.780]	22.540
	65	200	36.74	[49.332, 49.582]	49.625
	67	25	29.97	[25.486, 25.696]	22.982
	67	50	31.31	[46.196, 46.438]	44.481
	67	100	33.87	[90.634, 90.898]	89.733
	67	200	36.69	[190.673, 190.915]	190.198
	70	25	29.25	[108.154, 108.444]	104.035
	70	50	31.11	[194.318, 194.645]	191.700
	70	100	33.78	[369.696, 370.025]	368.188

Table 9: Hyperparameter search for output-wise subsample SDEdit on **CelebA**. The base grid sweeps re-noising steps $T \in \{1, 3, 5, 7, 9\}$, proportion of empirical (denoiser) generation $p \in \{55, 62, 65, 67, 70\}\%$, and subsample batch size $B \in \{25, 50, 100, 200\}$; at $T = 3$ we additionally refine B (to $\{60, 70\}$ for $p \in \{62, 65\}\%$ and to $\{80, 90, 120, 140\}$ for $p \in \{65, 67\}\%$), giving 112 configurations. FID measures sample quality (lower is better). PLD $[\varepsilon_\ell, \varepsilon_u]$ is the PLD interval under the sampling model of Section 4 (optimistic lower / pessimistic upper estimate, Google `dp_accounting`). Only ε_u is a certified bound. $\varepsilon_{\mathrm{sym}}$ is the (diagnostic) Edgeworth symmetric privacy loss; it is marked *invalid* where the reuse validity flag is false or a cumulant ratio $\rho \geq 1$ (diagnostics available only for the refined $T = 3$ rows).

Noise Steps	Prop. Empirical (%)	Subsample Batch Size	FID ↓	PLD $[\varepsilon_\ell, \varepsilon_u]$ ↓	$\varepsilon_{\mathbf{sym}}$ ↓
1	55	25	116.96	[0.000, 0.099]	0.099
	55	50	113.07	[0.000, 0.208]	0.208
	55	100	109.95	[0.185, 0.435]	0.435
	55	200	107.90	[0.666, 0.916]	0.915
	62	25	45.45	[0.516, 0.767]	0.563
	62	50	45.23	[1.179, 1.430]	1.150
	62	100	46.24	[2.308, 2.558]	2.361
	62	200	47.05	[4.654, 4.904]	4.912
	65	25	37.32	[3.623, 3.891]	*invalid* (ρ_4^Q=3.32)

Continued on next page

Table 9 continued

Noise Steps	Prop. Empirical (%)	Subsample Batch Size	FID ↓	PLD $[\varepsilon_\ell, \varepsilon_u]$ ↓	$\varepsilon_{\mathbf{sym}}$ ↓
	65	50	38.20	[5.249, 5.500]	*invalid* (ρ_4^Q=1.48)
	65	100	39.87	[8.482, 8.737]	7.147
	65	200	41.71	[15.522, 15.769]	14.552
	67	25	30.23	[13.360, 13.657]	*invalid* (ρ_4^Q=3.46)
	67	50	32.07	[20.517, 20.746]	*invalid* (ρ_4^Q=1.28)
	67	100	34.65	[34.120, 34.392]	30.085
	67	200	37.48	[61.908, 62.120]	58.631
	70	25	24.58	[52.702, 53.019]	*invalid* (ρ_4^P=2.26)
	70	50	27.01	[83.289, 83.470]	*invalid* (ρ_4^P=1.15)
	70	100	30.48	[140.112, 140.393]	132.911
	70	200	33.82	[251.533, 251.653]	245.964
3	55	25	35.16	[0.000, 0.099]	0.099
	55	50	35.78	[0.000, 0.208]	0.208
	55	100	37.42	[0.185, 0.435]	0.435
	55	200	39.30	[0.666, 0.916]	0.915
	62	25	25.56	[0.516, 0.767]	0.563
	62	50	27.57	[1.179, 1.430]	1.150
	62	60	28.36	[1.415, 1.666]	1.388
	62	70	28.85	[1.644, 1.893]	1.628
	62	100	30.31	[2.308, 2.558]	2.361
	62	200	33.01	[4.654, 4.904]	4.912
	65	25	21.95	[3.623, 3.891]	*invalid* (ρ_4^Q=3.32)
	65	50	24.34	[5.249, 5.500]	*invalid* (ρ_4^Q=1.48)
	65	60	25.01	[5.886, 6.129]	*invalid* (ρ_4^Q=1.19)
	65	70	25.67	[6.531, 6.767]	4.999
	65	80	26.30	[7.167, 7.415]	5.712
	65	90	26.81	[7.818, 8.072]	6.428
	65	100	27.25	[8.482, 8.737]	7.147
	65	120	27.99	[9.839, 10.090]	8.596
	65	140	28.72	[11.228, 11.472]	10.060
	65	200	30.05	[15.522, 15.769]	14.552
	67	25	17.94	[13.360, 13.657]	*invalid* (ρ_4^Q=3.46)
	67	50	20.39	[20.517, 20.746]	*invalid* (ρ_4^Q=1.28)
	67	80	22.50	[28.670, 28.950]	24.433
	67	90	23.16	[31.387, 31.670]	27.260
	67	100	23.67	[34.120, 34.392]	30.085
	67	120	24.63	[39.619, 39.854]	35.739
	67	140	25.35	[45.108, 45.350]	41.411
	67	200	26.93	[61.908, 62.120]	58.631
	70	25	14.70	[52.702, 53.019]	*invalid* (ρ_4^P=2.26)
	70	50	17.44	[83.289, 83.470]	*invalid* (ρ_4^P=1.15)
	70	100	20.97	[140.112, 140.393]	132.911
	70	200	24.54	[251.533, 251.653]	245.964
5	55	25	28.81	[0.000, 0.099]	0.099
	55	50	29.63	[0.000, 0.208]	0.208
	55	100	31.20	[0.185, 0.435]	0.435
	55	200	32.72	[0.666, 0.916]	0.915
	62	25	26.17	[0.516, 0.767]	0.563
	62	50	28.28	[1.179, 1.430]	1.150
	62	100	31.21	[2.308, 2.558]	2.361
	62	200	33.94	[4.654, 4.904]	4.912
	65	25	24.37	[3.623, 3.891]	*invalid* (ρ_4^Q=3.32)
	65	50	26.86	[5.249, 5.500]	*invalid* (ρ_4^Q=1.48)
	65	100	29.66	[8.482, 8.737]	7.147
	65	200	32.50	[15.522, 15.769]	14.552
	67	25	22.17	[13.360, 13.657]	*invalid* (ρ_4^Q=3.46)
	67	50	24.53	[20.517, 20.746]	*invalid* (ρ_4^Q=1.28)

Continued on next page

Table 9 continued

Noise Steps	Prop. Empirical (%)	Subsample Batch Size	FID ↓	PLD $[\varepsilon_\ell, \varepsilon_u]$ ↓	$\varepsilon_{\mathbf{sym}}$ ↓
	67	100	27.70	[34.120, 34.392]	30.085
	67	200	30.67	[61.908, 62.120]	58.631
	70	25	20.05	[52.702, 53.019]	*invalid* (ρ_4^P=2.26)
	70	50	22.64	[83.289, 83.470]	*invalid* (ρ_4^P=1.15)
	70	100	25.88	[140.112, 140.393]	132.911
	70	200	29.24	[251.533, 251.653]	245.964
7	55	25	29.25	[0.000, 0.099]	0.099
	55	50	29.67	[0.000, 0.208]	0.208
	55	100	30.73	[0.185, 0.435]	0.435
	55	200	31.74	[0.666, 0.916]	0.915
	62	25	26.42	[0.516, 0.767]	0.563
	62	50	28.15	[1.179, 1.430]	1.150
	62	100	30.47	[2.308, 2.558]	2.361
	62	200	32.79	[4.654, 4.904]	4.912
	65	25	25.73	[3.623, 3.891]	*invalid* (ρ_4^Q=3.32)
	65	50	27.96	[5.249, 5.500]	*invalid* (ρ_4^Q=1.48)
	65	100	30.43	[8.482, 8.737]	7.147
	65	200	33.00	[15.522, 15.769]	14.552
	67	25	25.02	[13.360, 13.657]	*invalid* (ρ_4^Q=3.46)
	67	50	27.37	[20.517, 20.746]	*invalid* (ρ_4^Q=1.28)
	67	100	30.17	[34.120, 34.392]	30.085
	67	200	32.95	[61.908, 62.120]	58.631
	70	25	23.51	[52.702, 53.019]	*invalid* (ρ_4^P=2.26)
	70	50	25.97	[83.289, 83.470]	*invalid* (ρ_4^P=1.15)
	70	100	29.19	[140.112, 140.393]	132.911
	70	200	32.23	[251.533, 251.653]	245.964
9	55	25	31.27	[0.000, 0.099]	0.099
	55	50	31.25	[0.000, 0.208]	0.208
	55	100	31.69	[0.185, 0.435]	0.435
	55	200	32.19	[0.666, 0.916]	0.915
	62	25	27.50	[0.516, 0.767]	0.563
	62	50	28.47	[1.179, 1.430]	1.150
	62	100	29.87	[2.308, 2.558]	2.361
	62	200	31.47	[4.654, 4.904]	4.912
	65	25	26.39	[3.623, 3.891]	*invalid* (ρ_4^Q=3.32)
	65	50	27.89	[5.249, 5.500]	*invalid* (ρ_4^Q=1.48)
	65	100	29.73	[8.482, 8.737]	7.147
	65	200	31.58	[15.522, 15.769]	14.552
	67	25	26.09	[13.360, 13.657]	*invalid* (ρ_4^Q=3.46)
	67	50	27.78	[20.517, 20.746]	*invalid* (ρ_4^Q=1.28)
	67	100	29.93	[34.120, 34.392]	30.085
	67	200	32.34	[61.908, 62.120]	58.631
	70	25	25.30	[52.702, 53.019]	*invalid* (ρ_4^P=2.26)
	70	50	27.54	[83.289, 83.470]	*invalid* (ρ_4^P=1.15)
	70	100	30.32	[140.112, 140.393]	132.911
	70	200	32.90	[251.533, 251.653]	245.964

Table 10: CelebA-64, $B = 100$, three public re-noising steps, $\delta = 10^{-6}$, $N_{\text{out}} = 50{,}000$. FID is mean $\pm$ standard deviation over three generation seeds.

Handoff	Edgeworth estimate	PLD lower	Certified PLD upper	FID
55%	0.43	0.19	0.44	37.39 ± 0.05
62%	2.36	2.31	2.56	30.45 ± 0.17
65%	7.14	8.48	8.74	27.31 ± 0.07
67%	30.09	34.12	34.39	23.75 ± 0.08
70%	132.91	140.11	140.39	20.98 ± 0.05

Handoff (%)	Edgeworth estimate	CLT estimate	Certified PLD upper
55	0.299	0.296	0.299
62	1.683	1.388	1.691
65	5.331	4.313	6.289
67	22.982	45.101	25.696
70	104.035	25,905.494	108.444

Table 11: Edgeworth and CLT estimates of ε against the certified PLD upper bound (CIFAR-10, $B = 25$, three re-noising steps, $\delta = 10^{-5}$). Neither approximation is a bound. The CLT estimate understates the certified value by 18% and 31% at the 62% and 65% blocks and overstates it by more than 200 times at the 70% block. The Edgeworth estimate stays within 1% for $\varepsilon \lesssim 2$ and understates it by 15%, 11%, and 4% at the 65%, 67%, and 70% blocks.

K Selection of the Privacy Parameters ε and δ

For a fixed sampler, Propositions 4.1 and C.1 determine a complete privacy profile $\delta_{\mathcal{M}}(\varepsilon)$: a practitioner picks a target δ_{target} and reports the smallest ε with $\delta_{\mathcal{M}}(\varepsilon) \leq \delta_{\text{target}}$, so δ is not tuned only through ε — the pair describes one fixed mechanism. Arbitrarily small ε and δ cannot coexist at fixed utility; improving both requires changing the mechanism (lower sampling rate, fewer private steps or released outputs, smaller clipping bound, more noise, or an earlier handoff).

We used $\delta = 10^{-5} \approx 1/n$ for CIFAR-10 to match common DP generative-model comparisons; $1/n^2 = 4 \times 10^{-10}$ is indeed stricter, and we report both in Table 12.

Table 12: Certified RDP ε for CIFAR-10 output-wise subsampled blocks at $\delta = 10^{-5}$ and at the stricter $\delta = 4 \times 10^{-10} = 1/n^2$ ($B = 25$, $N_{\text{out}} = 50{,}000$).

Configuration	RDP ε, $\delta = 10^{-5}$	RDP ε, $\delta = 4 \times 10^{-10}$
55%, $B = 25$	0.36	0.71
62%, $B = 25$	2.63	5.16
65%, $B = 25$	7.81	12.87

L Computing theoretical ASR

Following Triastcyn & Faltings (2020), we calculate the theoretical ASR as follows. Suppose an adversary observed an ϵ-DP release o and tried to distinguish between two candidate datasets $\mathcal{D}, \mathcal{D}'$ with a uniform prior $p(\mathcal{D}) = p(\mathcal{D}') = 0.5$. Without loss of generality, let $\mathcal{D}$ be the correct dataset, then the adversary will infer the probability $p(\mathcal{D}|o)$ as $p(\mathcal{D}|o) = \frac{p(o|\mathcal{D})p(\mathcal{D})}{p(o|\mathcal{D})p(\mathcal{D})+p(o|\mathcal{D}')p(\mathcal{D}')} \leq \frac{p(o|\mathcal{D})}{p(o|\mathcal{D})+\exp(-\epsilon)p(o|\mathcal{D})} = \frac{1}{1+\exp(-\epsilon)}$. For (ε, δ)-DP the bound becomes $\frac{e^{\varepsilon}+\delta}{1+e^{\varepsilon}}$, which at $\delta = 10^{-5}$ changes no entry of Table 13 at the reported precision. So, for $\epsilon = 1$ the upperbound on the membership inference success probability is 73.1%, for $\epsilon = 5$, it is 99.33%, and for $\epsilon = 10$, it is 99.995%. Furthermore, the success probability of MIAs typically similar or lower than the theoretical success rate as shown in (Lowy et al., 2024).

Table 13: Membership inference. Theoretical: ASR implied by our ε for the subsampled empirical sampler at rate q. Numerical: ASR of the attack of Duan et al. (2023) on a neural denoiser trained on all of CIFAR-10. Both stop at time t of a $T = 1000$ step discretization. The neural denoiser is not subsampled, so the theoretical rows are not a bound on it.

t	T	$\frac{3}{4}T$	$\frac{1}{2}T$	$\frac{1}{4}T$
Cosine similarity (Fig. 1)	1.000	0.999	0.956	0.845
Theoretical ASR at $q = 0.5$	0.5019	0.5204	0.6909	0.9999
Theoretical ASR at $q = 0.01$	0.5000	0.5002	0.5031	0.6927
Theoretical ASR at $q = 0.005$	0.5000	0.5001	0.5012	0.6371
Numerical ASR	0.5005 ± 0.0025	0.5001 ± 0.0011	0.5072 ± 0.0007	0.5096 ± 0.0026

M BROADER IMPACT STATEMENT

This work studies privacy leakage that can arise during diffusion-based sampling and provides tools for quantifying how individual training records may influence generated outputs. Its primary positive impact is to support more privacy-aware deployment of generative models, especially in settings where training data may contain sensitive or proprietary information. By shifting attention from only differentially private training to sampling-time privacy accounting, the proposed framework may help practitioners design generative pipelines that better balance privacy and utility, for example through subsampling, early stopping, or hybrid use of public denoisers. This can make formal privacy analysis more accessible for large-scale diffusion systems where fully private training remains costly or impractical.

At the same time, the work has limitations and possible risks. The formal guarantees are derived for an empirical denoiser with tractable sensitivity, while modern deployed systems typically use neural denoisers whose exact data sensitivity remains difficult to characterize. As a result, the analysis should not be interpreted as a complete privacy certificate for arbitrary diffusion models. If misused or overstated, sampling-time privacy estimates could give a false sense of security. Moreover, improving the privacy–utility tradeoff of generative models may also make it easier to deploy high-quality synthetic data systems in sensitive domains, where failures could still expose private information or amplify biases present in the underlying data.

We therefore view this work as a step toward, rather than a substitute for, end-to-end privacy evaluation. Responsible use should combine sampling-time accounting with empirical privacy audits, membership-inference evaluation, dataset governance, and clear reporting of assumptions. More broadly, our analysis highlights that privacy in generative modeling is shaped not only by training, but also by how samples are produced, released, and reused.