DIFFICULTY CALIBRATION ACROSS TIME FOR MEMBERSHIP INFERENCE IN DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern diffusion models are trained on massive datasets that often include copyrighted or private images. Membership inference attacks (MIAs) address this by testing whether a given image was in a model's training set. The standard approach scores an image by its reconstruction loss and flags low-loss images as members. This often fails because some images are simply easier to reconstruct than others, so a simple unseen image can score a lower loss than a complex training image. Difficulty calibration separates memorization from how difficult an image is to begin with. We show how to calibrate along two axes of time: training time and diffusion time. First, across training time, we introduce Reference-Model Calibration (RMC), which compares a fine-tuned model's loss against an earlier checkpoint. This cleanly isolates newly memorized data and gives near-perfect auditing, even without text captions. Since earlier checkpoints are not always available, we next present Temporal Self-Calibration (TSC), which calibrates across diffusion time using only a single model. Our key observation is that an image's difficulty persists across noise levels, whereas memorization is localized in time. By comparing the loss at a target timestep against neighboring timesteps from the same model, TSC attenuates image difficulty using the model's own trajectory. Empirically, our methods beat the state of the art by a wide margin, and unlike most existing attacks, generalize across different model parameterizations such as x-, v-, and ϵ-prediction models.

1 INTRODUCTION

Diffusion Models (DMs) (Song & Ermon, 2019; Ho et al., 2020; Song et al., 2020) are currently the method of choice for high-fidelity image generation (Rombach et al., 2022; Ramesh et al., 2022). As their success relies heavily on massive, uncurated web-scale datasets, these models often inadvertently absorb personal photographs, copyrighted artworks, and proprietary content. This indiscriminate data collection creates privacy and copyright risks, with recent studies (Carlini et al., 2023; Somepalli et al., 2023) demonstrating that DMs can memorize and reproduce specific training examples during generation. Consequently, it has become increasingly important to reliably audit the underlying training corpora of these generative models.

Membership Inference Attacks (MIAs) (Shokri et al., 2017) audit this risk by predicting whether a specific sample was part of the training set. While typically framed as a security vulnerability, MIAs can also enable creators to verify whether their proprietary content was used for training. These attacks exploit statistical differences in model behavior on members versus non-members (Yeom et al., 2018). Although well-studied for earlier generative models such as GANs (Goodfellow et al., 2020) and VAEs (Kingma & Welling, 2013), with several attacks developed for these settings (Hayes et al., 2017; Hilprecht et al., 2019; Chen et al., 2020), the iterative denoising process of diffusion models makes traditional MIAs considerably less effective (Duan et al., 2023), motivating approaches tailored specifically to the diffusion process.

Probably the most standard approach to MIAs is evaluating the models loss on the target examples, and predicting the low loss examples as members. Early approaches applied this to DMs by directly measuring the reconstruction error at a single, isolated timestep (Matsumoto et al., 2023). While followup work improved upon this (Duan et al., 2023; Kong et al., 2024), a fundamental obstacle limits the accuracy of all these approaches: There are images that are intrinsically easy that achieve

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.

low loss regardless of membership and vice versa. This difficulty factor confounds membership making the loss an unreliable membership metric.

To overcome this confound, we propose difficulty calibration across two time axes: *training time* and *diffusion time*. First, across training time, we introduce *Reference-Model Calibration (RMC)* to audit fine-tuned models. By comparing a sample's fine-tuned loss against the base checkpoint loss, RMC cleanly isolates newly memorized data from general reconstruction difficult. Note that this strategy is known for Large Language Models (Wang et al., 2024) but is under-explored in DMs. Its limitation is that prior checkpoints are frequently unavailable, and the approach is irrelevant for pretraining corpora. To provide an unconstrained alternative, we propose *Temporal Self-Calibration (TSC)*, which calibrates across diffusion time using only a single model. TSC leverages the insight that image difficulty persists across nearby noise levels, whereas the membership signal is highly localized in time. By evaluating the relative loss differences between a target timestep and its temporal neighbors, TSC uses the local trajectory as an internal baseline to suppress difficulty-induced variance. This internal calibration isolates the structural deviation of membership, successfully identifying a "difficult" member even when its absolute loss strictly exceeds that of an "easy" non-member.

We evaluate our methods across multiple model architectures, datasets, and training targets. While RMC provides state-of-the-art auditing for fine-tuned Stable Diffusion models, TSC serves as a highly powerful alternative for unconstrained settings. Across DDPM and Stable Diffusion evaluation settings, TSC consistently outperforms prior baselines. On CIFAR-10 (Krizhevsky et al., 2009), it doubles the performance of existing attacks to reach a 62.3% TPR@1%FPR. Furthermore, we emphasize TSC's robustness across diverse training objectives. While existing attacks are rigidly tailored to standard noise-prediction models and severely degrade on alternative parameterizations, TSC maintains highly accurate extraction capabilities. Finally, we analyze where the membership signal physically appears along the denoising trajectory, demonstrating that the effective leakage window is intrinsically tied to the underlying training objective.

Our contributions are:

- We propose *TSC*, a generalized method for robust membership inference. By evaluating relative loss differences across adjacent timesteps, TSC uses the local trajectory as an internal baseline to mitigate image difficulty and isolate membership signal.
- We introduce *RMC* for auditing fine-tuned models. To the best of our knowledge, we are the first to systematically analyze and demonstrate the effectiveness of prior-checkpoint calibration of single-timestep losses for membership inference in diffusion models.
- We comprehensively evaluate both methods, achieving state-of-the-art performance across diverse architectures and datasets. Notably, TSC consistently yields massive improvements over prior trajectory-based baselines, frequently doubling the true positive rate at strict false-positive thresholds.
- We provide a detailed analysis of how the effective denoising timesteps for membership inference shift across different training objectives, extending robust membership inference beyond the standard ϵ-prediction, ϵ-loss paradigm.

2 Related Work

Membership inference attacks. Membership inference attacks (MIAs), originally formalized by Shokri et al. (2017), aim to determine whether a specific data record was included in a target model's training set. MIAs are traditionally categorized by the adversary's capabilities: white-box attacks assume access to model parameters, activations, or gradients (Nasr et al., 2019), whereas black-box attacks operate under a more restrictive setting, relying exclusively on the model's observable outputs (Choquette-Choo et al., 2021). Gray-box attacks occupy an intermediate threat model, granting access to quantities such as loss values or intermediate model states without requiring gradient-level information. In diffusion-model auditing, this distinction is often methodological: attacks that query intermediate denoising states generally assume local access to the model while avoiding the substantially greater computational cost of full white-box analysis. Most such query-based MIAs are *threshold-based attacks*: they define a continuous membership score $\mathcal{S}_\theta(x)$, such as reconstruction loss or prediction error, for a target model θ and candidate sample x, and classify the sample as

a member when $\mathcal{S}_\theta(x) < \tau$ (Yeom et al., 2018; Carlini et al., 2022). The threshold τ can be empirically calibrated to optimize attack performance, often subject to a desired False Positive Rate (FPR). More sophisticated attacks additionally use shadow models to calibrate the score distribution (Shokri et al., 2017; Carlini et al., 2022).

MIAs on diffusion models. Auditing diffusion models requires analyzing the iterative denoising trajectory. While Carlini et al. (2022) achieved high precision with the Likelihood Ratio Attack (LiRA), their approach requires training numerous shadow models, making it computationally prohibitive. Consequently, efficient threshold-based attacks have become the standard. Matsumoto et al. (2023) introduced the Naive loss attack, which utilizes the diffusion training loss at specific timesteps as the membership score. To extract a more robust signal, Duan et al. (2023) proposed SecMI, which evaluates the reconstruction error along a deterministic trajectory via DDIM inversion, effectively isolating the membership signal from the high variance of stochastic forward noise. Kong et al. (2024) introduced the Proximal Initialization Attack (PIA), which initializes the forward process with a known deterministic noise vector near the clean data sample. PIA infers membership by evaluating the model's noise estimation against this explicit ground-truth trajectory, achieving competitive accuracy with a reduced query budget. Finally, for conditional text-to-image models, Zhai et al. (2024) proposed CLiD, which measures the discrepancy between conditional and unconditional diffusion losses to expose memorization specific to the text-image pair.

3 METHODOLOGY

3.1 PRELIMINARIES

Generative trajectories and log-SNR. Both diffusion (Ho et al., 2020; Song et al., 2020) and flow-matching (Lipman et al., 2022) models corrupt a clean sample x with Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ into noisy states $z_t = \alpha_t x + \sigma_t \epsilon$ over continuous time t. Diffusion defines (α_t, σ_t) via a variance schedule, whereas linear flow matching introduced by Liu et al. (2022) sets $\alpha_t = t$ and $\sigma_t = 1 - t$ for $t \in [0, 1]$, yielding $z_t = tx + (1 - t)\epsilon$ with target velocity $v = x - \epsilon$. Because native time variables differ across model families, we align all trajectories using the log-signal-to-noise ratio (log-SNR):

$$\lambda(t) = \log(\alpha_t^2/\sigma_t^2) \tag{1}$$

which simplifies to $\lambda(t) = 2\log(t/(1-t))$ for linear flow, mapping noise-dominated ($\lambda \to -\infty$) to data-dominated ($\lambda \to \infty$) states.

Prediction and loss parameterizations. A denoising network $f_\theta(z_t, t, c)$ (with optional conditioning c) predicts a native target $p \in \{x, v, \epsilon\}$. Because x, ϵ, and v are deterministically linked via z_t, any native prediction $\widehat{p}_\theta$ can be algebraically converted to evaluate the squared denoising error:

$$L_\theta(x, \lambda) = \|q - \widehat{q}_\theta(z_t, t, c)\|_2^2, \quad q \in \{x, v, \epsilon\} \tag{2}$$

all nine (p, q) combinations (Li & He, 2026) are detailed in Appendix A. Evaluating $L_\theta(x, \lambda)$ across λ for a fixed sample x yields its *denoising-loss trajectory*.

3.2 MOTIVATION

Most diffusion MIAs rely on a standard premise: models denoise training data more accurately than unseen data. However, absolute denoising loss varies wildly due to *image difficulty* factors like high-frequency details (Lian et al., 2025), complex layouts, or rare semantic concepts that make reconstruction harder. Consequently, an "easy" non-member often incurs a lower loss than a "difficult" member, deeply entangling the two populations (Fig. 1). Disentangling membership from this sample difficulty requires evaluating candidates against an image-specific reference rather than relying on absolute loss.

Calibration across training time. For models fine-tuned from a foundational base checkpoint, the original model provides a natural reference across *training time*. It establishes an accurate baseline of each image's difficulty *before* the audited fine-tuning stage. As shown for Stable Diffusion on MS-COCO (Lin et al., 2014) (Fig. 2), the losses produced by the two checkpoints are highly correlated. However, subtracting the pretrained loss to compute the fine-tuning residual reveals a stark

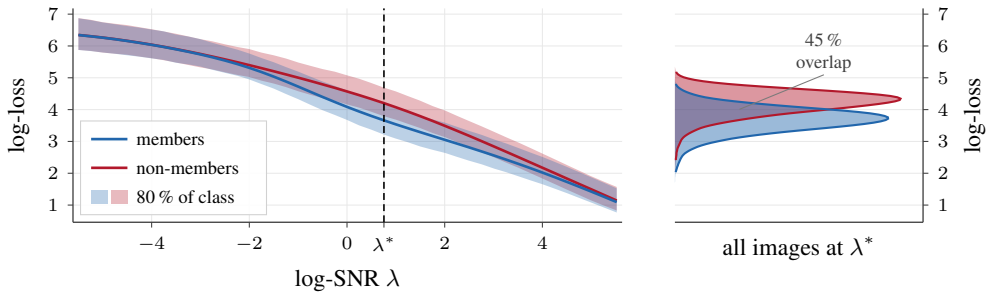

Figure 1: **Image difficulty confounds absolute-loss thresholds.** *Left:* CIFAR-10 DDPM log-loss trajectories for members (blue) and non-members (red) showing class means and 80% intervals. *Right:* Loss density at the optimal log-SNR ($\lambda^\star$). Even at maximal separability, the populations exhibit a massive 45% overlap, rendering absolute-loss thresholds ineffective.

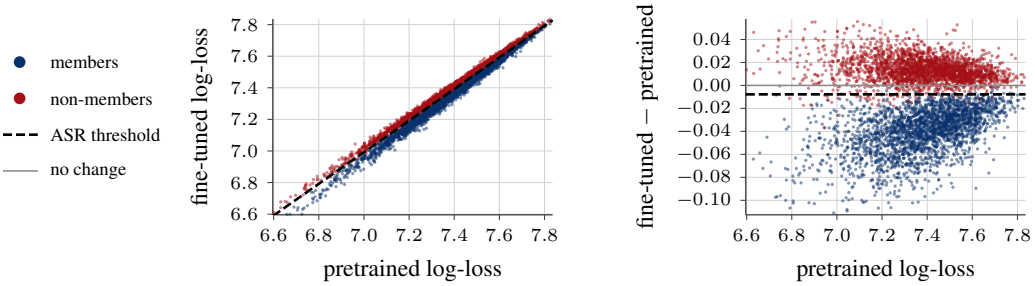

Figure 2: **RMC on MS-COCO.** Stable Diffusion fine-tuned on MS-COCO in realistic settings, uncaptioned. *Left:* Pretrained and fine-tuned log-losses are almost perfectly correlated (Pearson correlation $r = 0.989$). *Right:* Subtracting the pretrained loss removes this shared difficulty. Members shift below the ASR threshold (dashed ASR 98.6%), while non-members stay concentrated near 0.

separation: members systematically shift toward lower fine-tuned loss, while non-members remain concentrated near zero.

Table 1 demonstrates the empirical strength of this baseline against CLiD. By isolating the loss residual and thereby neutralizing image difficulty, a minimal two-query, uncaptioned reference evaluation heavily outperforms a 15-query, fully captioned, CLiD attack. Expanding to a comparable 20-query budget yields near-perfect auditing across all datasets. These results demonstrate that separating the membership signal from baseline difficulty is crucial for strict low-FPR auditing, directly motivating our reference-free temporal approach. Appendix E.5 reports the noise-level sweep.

Calibration across diffusion time. While a pretrained checkpoint provides an exceptionally strong baseline, assuming access to one is often unrealistic and irrelevant for auditing pretraining corpus. To achieve difficulty calibration without privileged external models, our primary method instead calibrates across *diffusion time*, leveraging the denoising trajectory itself as a self-contained internal reference. Because image difficulty generally evolves smoothly across nearby noise levels, adjacent timesteps provide an image-specific estimate of the local loss baseline. Membership, however, often manifests as a localized deviation from this baseline. As illustrated in Figure 3, comparing each loss against its local temporal reference largely cancels out overall image difficulty, successfully separating members from non-members based purely on trajectory shape.

3.3 Difficulty Calibration Across Time

To mitigate the difficulty confound observed in Section 3.2, we compare a candidate's loss against an image-specific reference. We hypothesize that a significant component of difficulty acts as a persistent multiplicative factor $H(x)$. If two loss evaluations satisfy $L_a(x) = H(x)\widetilde{L}_a(x)$ and $L_b(x) = H(x)\widetilde{L}_b(x)$, where $\widetilde{L}$ represents the intrinsic loss, their ratio cleanly isolates $\widetilde{L}$ by can-

Table 1: **RMC on realistic Stable Diffusion fine-tunes.** Reference calibration is evaluated without captions at $\lambda = 1$ log-SNR using either one or 10 paired noise draws. CLiD uses the ground-truth caption. Parentheses provide the number of model queries per candidate.

Method		Pokémon			MS COCO			Flickr		
		AUC	ASR	TPR@1%FPR	AUC	ASR	TPR@1%FPR	AUC	ASR	TPR@1%FPR
CLiD	($q = 15$)	96.98	91.83	64.75	95.55	88.72	44.04	95.22	88.16	47.14
RMC	($q = 2$)	99.44	96.51	90.63	99.88	98.52	96.60	99.80	98.14	96.86
	($q = 20$)	**99.73**	**97.84**	**96.64**	**99.94**	**99.04**	**98.52**	**99.96**	**99.50**	**99.82**

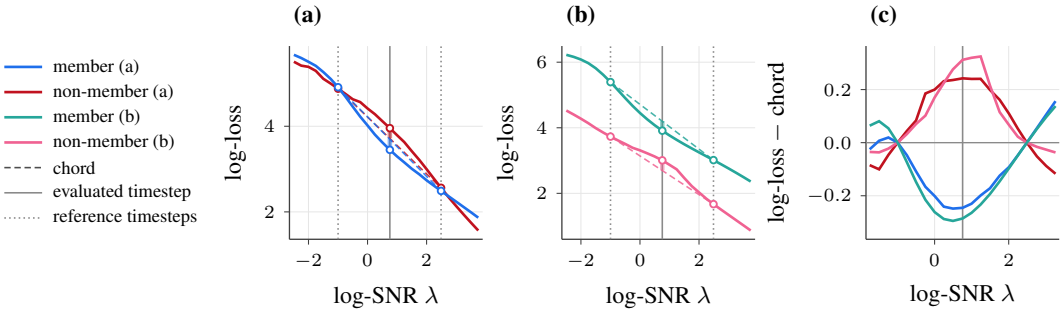

Figure 3: **Temporal self-calibration.** (a) A member/non-member pair where absolute loss correctly identifies the member. (b) A pair where absolute loss fails because the member is harder, yet a localized "dip" is still visible. (c) Subtracting the local temporal reference (the chord between the reference timesteps) removes the absolute difficulty offset. Both pairs collapse into a shared calibrated view, distinctly separating members from non-members by their trajectory shape.

celing $H(x)$. While idealized, Section 4.4 demonstrates this effectively reduces difficulty-induced variance in practice.

We perform this comparison in the logarithmic domain. Given a model θ, image x, log-SNR λ, and raw loss $L_\theta(x, \lambda)$ as described in Eq. 2, we define the stabilized log-loss:

$$\ell_\theta(x, \lambda) = \log(L_\theta(x, \lambda) + \delta)\,, \tag{3}$$

where $\delta > 0$ is a small stability constant. Subtracting two such log-losses effectively computes their log-ratio, canceling $H(x)$. We instantiate this relative calibration across two time axes: *training time* (using a pretrained base checkpoint) and *diffusion time* (using the image's own trajectory).

3.3.1 Reference-Model Calibration

For audited fine-tuned models (θ_{FT}) with an available pretrained checkpoint (θ_{pre}), the base model provides a natural estimate of an image's intrinsic difficulty prior to membership exposure. Subtracting this baseline cleanly isolates the behavioral shift caused by fine-tuning. For a fixed log-SNR λ, we define the reference-calibrated residual:

$$r_{\text{ref}}(x, \lambda) = \ell_{\theta_{\text{FT}}}(x, \lambda) - \ell_{\theta_{\text{pre}}}(x, \lambda). \tag{4}$$

As fine-tuned members incur a lower loss on θ_{FT} relative to the pretrained baseline, they systematically produce a more negative residual. We therefore use $r_{\text{ref}}(x, \lambda)$ directly as our membership score.

3.3.2 Temporal Self-Calibration

To calibrate across *diffusion time*, TSC uses a candidate's own loss trajectory as an internal reference. Image difficulty tends to persist across noise levels, affecting the overall position of an image's log-loss trajectory. However, membership advantage varies along the trajectory. As illustrated in Figure 4, members tend to exhibit a greater reduction in log-loss along a certain window in the process, relative to the surrounding trajectory, than non-members.

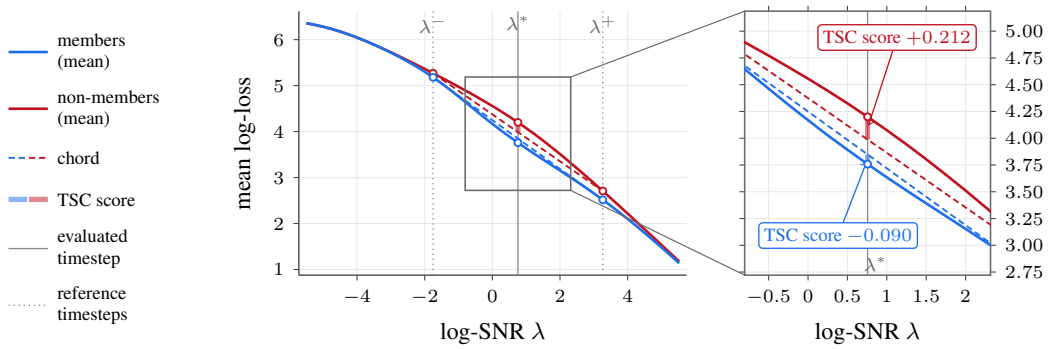

Figure 4: **TSC on STL-10-U.** Mean member and non-member log-loss trajectories (left) and a close-up of the selected window (right). The chords connect the two context points for each population, and the vertical bars at λ^* show the residual measured by the temporal score.

Let λ^* denote a selected query point (e.g., the region of maximum separation), and let $\lambda^- = \lambda^* - \Delta$ and $\lambda^+ = \lambda^* + \Delta$ denote two symmetric context points. We define the temporal residual as

$$r_{\text{temp}}(x) = \ell_\theta(x, \lambda^*) - \frac{1}{2}\left(\ell_\theta(x, \lambda^-) + \ell_\theta(x, \lambda^+)\right). \tag{5}$$

Geometrically, this three-point symmetric stencil measures the deviation of the log-loss at λ^* relative to the linear chord that connects its two neighbors.

Under the multiplicative difficulty model, the stencil weights in Eq. 5 sum to zero, canceling the shared additive offset $\log H(x)$. Moreover, placing the context points symmetrically around λ^* cancels linear trends and any trajectory components antisymmetric about λ^* even at large window widths Δ, eliminating residual slope variations that a single reference would retain (see Appendix C and Section 4.4).

Crucially, because membership leakage is a macro-scale phenomenon, we avoid narrow windows that merely measure local curvature. Instead, we position λ^* at the intermediate memorization peak and set a wide span 2Δ so λ^- and λ^+ fall in lower-signal regions that anchor the image's baseline trajectory. For members, memorization at λ^* pulls the log-loss further downward relative to the linear chord connecting λ^- and λ^+, yielding a systematically more negative residual $r_{\text{temp}}(x)$ than for non-members.

4 Experiments

4.1 Experiment Setup

Models and datasets. For standard image synthesis, we evaluate DDPMs (Ho et al., 2020) on CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009), STL-10-U (Coates et al., 2011), and Tiny ImageNet (Deng et al., 2009), as well as DDPM++ backbone (Song et al., 2020) on CIFAR-100. For text-to-image generation, we evaluate Stable Diffusion v1.4 (Rombach et al., 2022) fine-tuned on the Pokémon (Pinkney, 2022), MS-COCO (Lin et al., 2014), and Flickr30k (Young et al., 2014) datasets. Complete model configurations and dataset splits are detailed in Appendix B.

Baselines and evaluation metrics. We compare TSC against established MIAs for DMs: Naive (Matsumoto et al., 2023), SecMI (Duan et al., 2023), PIA, and PIAN (Kong et al., 2024). We measure performance across all methods using Attack Success Rate (ASR), AUC, and True Positive Rate at 1% False Positive Rate (TPR@1%FPR).

Implementation details. Hyperparameters for both TSC (the optimal center $\lambda^\star$) and timestep-dependent baselines (Naive, PIA) are tuned by maximizing AUC on a disjoint shadow data split. As analyzed in our results, these optimal coordinates naturally shift depending on the model's loss weighting. For TSC, we fix the temporal window at $2\Delta = 5$ log-SNR. To balance efficiency, we

Table 2: **Membership inference on DDPM benchmarks.** Results reported as percentages. Parentheses give the number of model queries per candidate. SecMI follows its official protocol.

Method		CIFAR-10			Tiny ImageNet			STL-10-U		
		AUC	ASR	TPR@1%FPR	AUC	ASR	TPR@1%FPR	AUC	ASR	TPR@1%FPR
Naive	($q = 20$)	85.39	78.21	7.35	64.37	60.52	2.93	64.15	60.50	2.51
SecMI	($q = 12$)	88.24	80.91	10.20	66.37	61.88	2.91	68.12	63.33	3.03
PIA	($q = 2$)	<u>91.90</u>	<u>84.88</u>	25.80	<u>68.99</u>	<u>64.25</u>	4.43	<u>69.46</u>	<u>64.51</u>	4.08
PIAN	($q = 2$)	89.26	83.32	<u>30.90</u>	67.04	63.44	<u>6.10</u>	66.72	63.36	<u>4.32</u>
TSC	($q = 15$)	**95.91**	**90.15**	**62.31**	**78.63**	**72.18**	**12.27**	**77.84**	**71.29**	**9.57**

Table 3: **Membership inference on Stable Diffusion fine-tunes (uncaptioned).** SecMI denotes its LDM variant. PIA, is evaluated both over the original paper's default timestep ($t = 500$) and an average optimal timestep over shadow models.

Method		Pokémon			MS COCO			Flickr		
		AUC	ASR	TPR@1%FPR	AUC	ASR	TPR@1%FPR	AUC	ASR	TPR@1%FPR
Naive	($q = 10$)	84.84	77.43	28.37	76.61	69.92	7.16	77.96	71.66	10.08
SecMI	($q = 12$)	83.59	76.47	15.14	<u>78.33</u>	<u>71.46</u>	6.84	<u>81.14</u>	<u>74.24</u>	7.40
PIA ($t = 500$)	($q = 2$)	69.86	66.13	19.71	54.08	53.96	2.76	54.73	53.24	1.96
PIA (optimal t)	($q = 2$)	**94.50**	**87.88**	**61.30**	75.61	70.90	<u>17.36</u>	74.18	69.58	<u>17.12</u>
PIAN	($q = 2$)	53.34	54.15	1.92	69.20	64.08	13.88	69.15	65.10	12.40
TSC	($q = 15$)	<u>91.37</u>	<u>84.03</u>	<u>55.77</u>	**85.55**	**78.42**	**29.16**	**84.60**	**77.98**	**30.28**

allocate a proportionate query budget that ensures stable evaluation. However, as shown in Fig.5, the attack performance can improve further with higher query budget.

4.2 Temporal Self-Calibration Evaluations and Analysis

Table 2 presents the performance of our method TSC against the baselines on standard image synthesis models. TSC consistently achieves state-of-the-art performance across all datasets. While it provides solid improvements in overall AUC and ASR, its most significant advantage emerges in the high-confidence regime. On CIFAR-10, TSC achieves a TPR@1%FPR of 62.31%, doubling the performance of the strongest baseline (PIAN at 30.90%). This trend persists across the more complex Tiny ImageNet and STL-10-U datasets, demonstrating that our geometric measurement of the loss trajectory effectively isolates members while bounding false positives.

For text-to-image models (Table 3), we evaluate Stable Diffusion under a realistic uncaptioned threat model. Because training prompts are typically unknown to an auditor, omitting ground-truth captions prevents artificially inflating attack success via prompt-specific overfitting, directly isolating the inherent image-level membership signal.

Under this setting, TSC outperforms all baselines across every metric on MS-COCO and Flickr30k. The exception is the Pokémon fine-tune, where PIA prevails. We attribute this to the dataset's limited scale and visual uniformity. Such low diversity naturally minimizes the variance in image difficulty, allowing PIA's proximal measurement to perform exceptionally well without requiring relative calibration. Furthermore, our shadow-model tuning shifts PIA's optimal timestep from its official default of $t = 500$ (Kong et al., 2024) down to $t \approx$ 250-300. By evaluating PIA at these tuned coordinates, our reported baselines are notably stronger than standard static evaluations.

4.3 Impact of Prediction Parameterization and Training Target

Membership leakage varies along the trajectory, making timestep selection crucial. To understand how the training objective influences the informative region, we express the losses in a common velocity coordinate.

Table 4: **Robustness across prediction and loss parameterizations.** TPR@1%FPR (%) for unaugmented CIFAR-100 DDPM++ FM models. The selected-$\lambda^\star$ tuple lists the shared operating log-SNR under x-, v-, and ϵ-losses. SecMI follows its default protocol.

Method		Selected $\lambda^\star$ ($x/v/\epsilon$ loss)	x-loss			v-loss			ϵ-loss		
			x-pred	v-pred	ϵ-pred	x-pred	v-pred	ϵ-pred	x-pred	v-pred	ϵ-pred
Naive	($q = 20$)	$-0.75/0.25/1.50$	6.48	6.05	6.74	34.54	34.28	34.56	27.52	29.78	28.29
PIA	($q = 2$)	$-0.75/0.25/1.75$	2.27	4.07	6.22	1.01	52.48	54.84	31.69	74.61	75.85
PIAN	($q = 2$)	$-0.50/0.50/1.75$	2.41	6.00	5.69	1.06	56.68	58.36	72.11	77.77	76.73
SecMI	($q = 12$)	default	0.94	1.10	1.19	5.55	5.57	5.26	29.25	35.40	30.82
TSC	($q = 15$)	$-0.50/0.50/1.75$	**30.28**	**28.74**	**31.26**	**81.31**	**80.56**	**80.68**	**91.22**	**93.22**	**92.75**

Under linear flow, $x = z_t + (1-t)v$ and $\epsilon = z_t - tv$. For a fixed native head $p \in \{x, v, \epsilon\}$ and $t \in (0, 1)$, a converted velocity error $\delta_v = \widehat{v}_\theta^{(p)} - v$ induces the errors

$$\delta_x = (1-t)\delta_v, \qquad \delta_\epsilon = -t\delta_v.$$

Squaring these residuals gives $\mathcal{L}_{p,s}(t) = w_s(t)\|\delta_v\|_2^2$, where $(w_x, w_v, w_\epsilon) = ((1-t)^2, 1, t^2)$ depends only on the training-loss target $s \in \{x, v, \epsilon\}$.

Lemma 1 (Temporal Weight Ordering)**.** For a common non-degenerate timestep distribution ρ on $(0, 1)$, the explicit-weight barycenters $c_s \triangleq \mathbb{E}_\rho[T w_s(T)]/\mathbb{E}_\rho[w_s(T)]$ satisfy $c_x < c_v < c_\epsilon$. Under uniform sampling, $(c_x, c_v, c_\epsilon) = (1/4, 1/2, 3/4)$. Proof in Appendix D.2.

In this coordinate, x-loss emphasizes noisy timesteps and ϵ-loss emphasizes cleaner timesteps, while v-loss adds no temporal weighting beyond ρ. This ordering concerns explicit weights, it does not determine leakage peaks, which also depends on learned errors and optimization dynamics.

We examine whether the measured leakage follows the same ordering across nine separately trained CIFAR-100 models spanning all (p, s) combinations. Single-timestep AUC sweeps (Fig. 6, Appendix D.3) show that peak leakage shifts toward higher log-SNR as the training target changes from x to v to ϵ, whereas changing the native head p has little effect on the peak location.

Table 4 evaluates attacks across all nine models. Consistent with the observed ordering, TSC's center increases from $\lambda^\star = -0.50$ (x-loss) to 0.50 (v-loss) and 1.75 (ϵ-loss). TSC achieves the highest TPR@1%FPR in every configuration, improving over the strongest baseline by 23.84 percentage points on average. Under x-loss, baselines remain below 7% TPR while TSC reaches 29–31%. These results support adapting the operating region to the training objective and demonstrate TSC's effectiveness across the tested parameterizations.

4.4 Ablations

Score domain and query budget. We ablate the loss domain and query budget q (Fig. 5). Evaluating TSC on logarithmic rather than raw loss reveals a stark contrast: log-loss TSC reaches nearly 26% TPR@1%FPR, while raw-loss TSC falls below the Naive baseline. This strengthens our hypothesis that sample difficulty acts as a multiplicative confounder. The log transformation converts this into an additive constant, allowing TSC to effectively cancel it out.

Furthermore, as queries increase from $q = 3$ to $q = 45$, log-loss TSC scales efficiently. Additional noise draws at the same three query levels improve estimation of the temporal contrast. In this experiment, the Naive attack remains near 4%, gaining little from the additional loss evaluations.

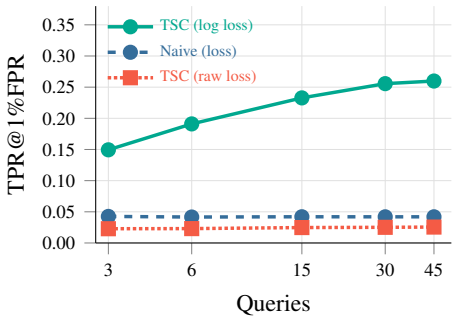

Figure 5: **Score domain and query budget.** Mean TPR@1%FPR over 8 DDPM checkpoints at matched query budgets. Both TSC scores use the same temporal stencil.

Filter selection. At $\lambda^\star = 0.75$, we compare endpoint references with evenly spaced, Gaussian, and reverse-window filters, the latter assigning greater weight to distant levels. Here, k denotes the number of sampled levels on each side of the center, giving $2k+1$ queried levels in total. TSC uses $k = 1$, with references at $\lambda^\star \pm \Delta$. Definitions and search grids appear in Appendix B.1.

Table 5: **Temporal reference filters.** Mean AUC and TPR@1%FPR across four datasets trained over DDPM settings. The full Gaussian uses 45 levels and is unavailable at $q = 15$.

		$q = 15$		$q = 45$	
Filter	Setting	AUC	TPR	AUC	TPR
TSC (endpoints)	$\Delta = 2.5,\ k = 1$	**87.06**	**35.30**	**88.11**	**39.26**
Even	$\Delta = 2.5,\ k = 2$	83.96	26.43	86.84	34.84
Gaussian (full)	$\sigma = 3$	–	–	81.21	20.57
Gaussian (window)	$k = 7,\ \sigma = 3$	71.84	5.72	79.53	11.28
Reverse window	$k = 7,\ \sigma = 0.5$	80.58	16.26	85.65	27.95

Endpoints achieve the highest mean AUC and TPR@1%FPR at both query budgets (Table 5). The reverse window improves over the center-weighted Gaussian window but remains below endpoints. At $q = 45$, repeated noise draws at fixed endpoints outperform the tested alternatives using additional temporal levels, supporting concentrating queries on the outer references.

Reference symmetry. At $\lambda^\star = 0.75$ and $q = 15$, we compare our symmetric stencil against single-sided references and unequal windows spanning $a + b = 5$. To maintain a fixed query budget, single-reference scores receive proportionally more draws per evaluated point. As shown in Table 6, symmetry achieves the highest mean TPR@1%FPR, outperforming all one-sided and asymmetric baselines. While an equal-weight $(1.5, 3.5)$ window yields a slightly higher overall AUC, the symmetric contrast strictly dominates in the low-FPR regime - the critical operating requirement for practical auditing. Furthermore, interpolation weighting fails to consistently improve asymmetric windows, empirically validating our symmetric default. Full theoretical diagnostics and per-dataset results are provided in Appendix C.1.

Table 6: **Temporal reference symmetry.** At $q = 15$, mean AUC and TPR@1%FPR across four datasets trained over DDPM settings. (a, b) denotes nominal log-SNR offsets.

Reference	(a, b)	AUC	TPR
Symmetric	$(2.5, 2.5)$	87.06	**35.30**
Left only	$(2.5, -)$	78.15	23.52
Right only	$(-, 2.5)$	82.13	6.47
Unequal	$(1.5, 3.5)$	**88.77**	30.94
Unequal	$(3.5, 1.5)$	80.64	26.56
Weighted	$(1.5, 3.5)$	86.24	29.11
Weighted	$(3.5, 1.5)$	83.62	26.58

5 Conclusion

We introduced difficulty calibration across two time axes to mitigate the image-difficulty confound in diffusion MIAs. *Reference-Model Calibration (RMC)* calibrates across *training time* using a pretrained checkpoint, while *Temporal Self-Calibration (TSC)* calibrates across *diffusion time* using the image's own denoising trajectory. Experiments across diverse datasets, architectures, and training objectives demonstrate substantial gains in strict low-FPR membership detection. RMC enables near-perfect uncaptioned auditing of fine-tuned data, and TSC frequently doubles the TPR@1%FPR of the strongest trajectory-based baselines in fully unconstrained settings.

Limitations and future work. Although our methods use only forward predictions and do not require gradients, they assume access to denoising outputs for candidate images at selected noise levels. This access may be unavailable through APIs that expose only final generated images, limiting direct application in such settings.

RMC additionally assumes access to the checkpoint from which the audited model was fine-tuned. Its applicability is therefore restricted when this checkpoint is unavailable or the model's training lineage is unknown. Future work could investigate whether related public checkpoints or ensembles of reference models can provide effective difficulty calibration in these cases.

RMC measures loss changes between checkpoints and does not separately identify membership in the reference model's pretraining corpus. Studying overlap between pretraining and fine-tuning data, and extending calibration across successive checkpoints, could clarify how membership evidence evolves across training stages.

AI USE STATEMENT

The proposed TSC and RMC methods were conceived and designed by the authors. We used generative AI tools to help identify relevant prior work, draft and polish the manuscript, refine the mathematical analysis and proofs, and assist with experimental implementation and execution. AI-generated code used in the experiments was reviewed and verified by both authors. Both authors take responsibility for the manuscript's content, methods, and reported results.

ETHICS STATEMENT

Membership inference has both privacy risks and legitimate auditing uses. Improved attacks could be misused to infer an individual's participation in sensitive training data. At the same time, these methods can help individuals investigate whether their personal data were used to train or fine-tune diffusion models, and can support audits of candidate training-data sources for publicly released models, including those distributed through platforms such as Hugging Face. Such audits can help assess claims about dataset use and improve transparency around model provenance. Membership scores provide statistical evidence about candidate examples: a negative result does not prove that an example was excluded from training, and an audit does not identify a model's complete training dataset. We advocate using these methods for authorized privacy auditing and accountability, with conclusions that account for false positives and false negatives.

REPRODUCIBILITY STATEMENT

We intend to publicly release the code needed to reproduce our experiments upon acceptance. The calibration methods are described in Section 3.3. The experimental settings and evaluation protocols are provided in Section 4.1 and Appendix B. The mathematical derivations appear in Appendices C and D, with additional results in Appendix E.

REFERENCES

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE symposium on security and privacy (SP)*, pp. 1897–1914. IEEE, 2022.

Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX security symposium (USENIX Security 23)*, pp. 5253–5270, 2023.

Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. Gan-leaks: A taxonomy of membership inference attacks against generative models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pp. 343–362, 2020.

Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *International conference on machine learning*, pp. 1964–1974. PMLR, 2021.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Jinhao Duan, Fei Kong, Shiqi Wang, Xiaoshuang Shi, and Kaidi Xu. Are diffusion models vulnerable to membership inference attacks? In *International Conference on Machine Learning*, pp. 8717–8730. PMLR, 2023.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan: Membership inference attacks against generative models. *arXiv preprint arXiv:1705.07663*, 2017.

Benjamin Hilprecht, Martin Härterich, and Daniel Bernau. Monte carlo and reconstruction membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

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

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Fei Kong, Jinhao Duan, Heng Tao Shen, Xiaoshuang Shi, Xiaofeng Zhu, Kaidi Xu, et al. An efficient membership inference attack for the diffusion model by proximal initialization. In *International Conference on Learning Representations*, volume 2024, pp. 57876–57896, 2024.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36115–36125, 2026.

Puwei Lian, Yujun Cai, Songze Li, and Bingkun Bao. Enhancing membership inference attacks on diffusion models from a frequency-domain perspective. *arXiv preprint arXiv:2505.20955*, 2025.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

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

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

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

Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*, pp. 739–753. IEEE, 2019.

Justin N. M. Pinkney. Pokemon blip captions. `https://huggingface.co/datasets/lambdalabs/pokemon-blip-captions/`, 2022.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18. IEEE, 2017.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6048–6058. IEEE, 2023.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

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.

Jeffrey G Wang, Jason Wang, Marvin Li, and Seth Neel. Pandora's white-box: Precise training data detection and extraction in large language models. *arXiv preprint arXiv:2402.17012*, 2024.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pp. 268–282. IEEE, 2018.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the association for computational linguistics*, 2:67–78, 2014.

Shengfang Zhai, Huanran Chen, Yinpeng Dong, Jiajun Li, Qingni Shen, Yansong Gao, Hang Su, and Yang Liu. Membership inference on text-to-image diffusion models via conditional likelihood discrepancy. *Advances in Neural Information Processing Systems*, 37:74122–74146, 2024.

A Prediction and Loss Parameterizations

As discussed in Section 3.1, a diffusion or flow-matching model can be parameterized to predict the clean data x, the noise ϵ, or the velocity v, and its training objective can be evaluated in any of these three spaces. Because these quantities are deterministically linked by the forward process, a prediction in one space can be algebraically converted to any other space before computing the loss.

Table 7 details the exact conversion equations for all nine combinations under the linear interpolation schedule $z_t = (1-t)\epsilon + tx$, following Li & He (2026).

Table 7: Prediction and loss parameterizations for the linear flow schedule. Columns specify the direct network prediction space, while rows specify the space in which its squared error is measured.

	x-prediction $x_\theta := f_\theta(z_t, t, c)$	**ϵ-prediction** $\epsilon_\theta := f_\theta(z_t, t, c)$	**v-prediction** $v_\theta := f_\theta(z_t, t, c)$
x-loss: $\mathbb{E}\|x_\theta - x\|_2^2$	x_θ	$x_\theta = \frac{z_t - (1-t)\epsilon_\theta}{t}$	$x_\theta = z_t + (1-t)v_\theta$
ϵ-loss: $\mathbb{E}\|\epsilon_\theta - \epsilon\|_2^2$	$\epsilon_\theta = \frac{z_t - tx_\theta}{1-t}$	ϵ_θ	$\epsilon_\theta = z_t - tv_\theta$
v-loss: $\mathbb{E}\|v_\theta - v\|_2^2$	$v_\theta = \frac{x_\theta - z_t}{1-t}$	$v_\theta = \frac{z_t - \epsilon_\theta}{t}$	v_θ

B Experimental Details

Common protocol. For all membership-inference experiments, each target model is trained only on a designated member split, and attacks are evaluated by distinguishing this member split from a disjoint hold-out split. Unless stated otherwise, images are normalized to $[-1, 1]$, models are trained with Adam, and we evaluate the final EMA checkpoint.

DDPM training. For the unconditional DDPM experiments, we use a UNet backbone with base width 128, channel multipliers $(1, 2, 2, 2)$, two residual blocks per resolution, dropout 0.1, and attention at the 16×16 resolution. We train ϵ-prediction DDPMs with $T = 1000$ diffusion steps, a linear noise schedule from $\beta_1 = 10^{-4}$ to $\beta_T = 0.02$, fixed-large variance, learning rate $2 \cdot 10^{-4}$, gradient clipping at 1.0, EMA decay 0.9999, and batch size 128. All three datasets below use random horizontal flips.

Table 8: DDPM training settings. All runs use 32×32 resolution, batch size 128, linear noise schedule, ϵ-prediction, and random horizontal flips.

Dataset	Resolution	Member	Hold-out	Aug.	Steps	Epochs
CIFAR-10	32×32	25k	25k	HFlip	800k	≈ 4k
CIFAR-100	32×32	25k	25k	HFlip	800k	≈ 4k
STL-10-U	32×32	50k	50k	HFlip	1.6M	≈ 4k
Tiny ImageNet	32×32	50k	50k	HFlip	1.6M	≈ 4k

Stable Diffusion fine-tuning. For text-to-image experiments, we fine-tune Stable Diffusion v1.4 at 512×512 resolution. We fine-tune the UNet while keeping the VAE and text encoder fixed, use mixed precision, gradient checkpointing, EMA, a constant learning-rate schedule, no warmup, gradient clipping at 1.0, batch size 1, gradient accumulation 4, and learning rate 10^{-5}. Unless stated otherwise, all runs use random cropping.

Caption data. For Pokémon, we use image–caption pairs from the Pokémon BLIP-caption dataset. For MS-COCO and Flickr30k, we use the corresponding image-caption annotations from the constructed train/hold-out splits. Unless stated otherwise, each image is paired with its staged caption metadata and no caption dropout is applied.

Table 9: Stable Diffusion v1.4 overfitting-setting fine-tuning details (Table 3) settings.

Dataset	Train	Hold-out	Steps	Aug.
Pokémon	416	417	7.5k	Crop
MS-COCO	2.5k	2.5k	150k	Crop
Flickr30k	2.5k	2.5k	150k	Crop

Table 10: Stable Diffusion v1.4 real-world-setting fine-tuning details (Table 1) settings.

Dataset	Train	Hold-out	Steps	Aug.
Pokémon	416	417	7.5k	Crop+HFlip
MS-COCO	2.5k	2.5k	50k	Crop+HFlip
Flickr30k	10k	10k	200k	Crop+HFlip

DDPM++ In addition to the DDPM models, we evaluate a DDPM++ backbone implemented with EDM (Karras et al., 2022) base code ddpmpp over CIFAR-100 and switching the loss for the 9 different linear flow matching prediction/loss pairs. This model is trained over the same settings as stated in Table 8 except no data augmentation and for 2000 epochs. We use this setting to isolate the effect of the native prediction head and the training-loss target. Specifically, the architecture, data split, optimizer, and training budget are fixed, while we vary the prediction/loss variants reported in the corresponding tables.

B.1 Temporal-Filter Selection

The filter ablation evaluates 45 log-SNR levels from -5.5 to 5.5 at intervals of 0.25, with the center fixed at $\lambda^\star = 0.75$. The endpoint stencil is fixed at $\Delta = 2.5$, $k = 1$, matching the main DDPM experiments. Even filters search $\Delta \in \{0.5, 1.0, \dots, 4.0\}$ and use $k \in \{2, \dots, 7\}$. Gaussian bandwidths are selected from $\sigma \in \{0.5, 1.0, 1.5, 2.0, 3.0\}$. Windowed Gaussian filters use $k \in \{1, \dots, 7\}$ nearest levels per side. Reverse windows use the same support construction with weights increasing away from the center, and search $k \in \{2, \dots, 7\}$.

All restricted supports include the central level and therefore contain $m = 2k + 1$ levels. Each support level is evaluated using $\lfloor q/m \rfloor$ independent noise draws. The even and windowed Gaussian families are selected under a budget of $q = 15$. The full Gaussian necessarily uses all 45 levels and is selected at $q = 45$. For each non-TSC family, one configuration is selected by mean AUC across the four first-half shadow models. The membership-score direction is fixed so lower temporal residuals are member-like. Selected supports and weights are reused on the matching held-out 4K checkpoints at both query budgets. Results are macro-averaged over CIFAR-10, CIFAR-100, STL-10-U, and Tiny ImageNet with horizontal flips.

C Temporal Reference Symmetry

We distinguish the removal of a shared loss level from the additional effect of symmetric reference points. The identities below hold over any finite window. They require neither a narrow-window approximation nor the interpretation of the membership feature as pointwise curvature. We define

$$\ell_x(u) = \ell_\theta(x, \lambda^\star + u) \tag{6}$$

with the stabilized log-loss defined in Eq. equation 3.

Shared difficulty and reflection symmetry. Under the idealized factorization $L_\theta(x, \lambda) = H(x)\widetilde{L}_\theta(x, \lambda)$, the unstabilized log-loss contains the additive offset $\log H(x)$. Every contrast whose coefficients sum to zero removes this offset, including one-sided and asymmetric contrasts. With the numerical stabilizer $\delta > 0$, this correspondence to multiplicative cancellation is approximate when δ is negligible relative to all evaluated losses. Scalar difficulty alone therefore does not single out symmetry.

For any profile, define its even and odd components about the query point:

$$E_x(u) = \frac{\ell_x(u) + \ell_x(-u)}{2}, \qquad O_x(u) = \frac{\ell_x(u) - \ell_x(-u)}{2}. \tag{7}$$

The symmetric temporal score is exactly

$$S_x = \ell_x(0) - E_x(\Delta). \tag{8}$$

Consequently, adding any sample-dependent offset a_x or odd function $q_x(-u) = -q_x(u)$ to $\ell_x(u)$ leaves S_x unchanged. This removes antisymmetric profile variation over the entire window, including nonlinear variation. A deterministic trajectory component shared by all images contributes only a common score offset and does not affect rankings. The relevant nuisance is variation between images.

Relation to one-sided references. Define $S_{x,+} = \ell_x(0) - \ell_x(\Delta)$ and $S_{x,-} = \ell_x(0) - \ell_x(-\Delta)$. Then

$$S_{x,+} = S_x - O_x(\Delta), \qquad S_{x,-} = S_x + O_x(\Delta). \tag{9}$$

Thus each one-sided contrast includes the same centered deviation and an additional odd component with opposite signs. Let $M \in \{0, 1\}$ denote membership, abbreviate $S = S_x$, $S_\pm = S_{x,\pm}$, and $O = O_x(\Delta)$, and write Var_m and Cov_m for variance and covariance conditional on $M = m$. Assuming finite second moments,

$$\mathrm{Var}_m(S_\pm) = \mathrm{Var}_m(S) + \mathrm{Var}_m(O) \mp 2\,\mathrm{Cov}_m(S, O), \tag{10}$$

$$\frac{\mathrm{Var}_m(S_+) + \mathrm{Var}_m(S_-)}{2} = \mathrm{Var}_m(S) + \mathrm{Var}_m(O). \tag{11}$$

If O has equal conditional means for members and non-members, both one-sided contrasts have the same mean membership gap as S, while their average conditional variance is larger whenever O varies. Both individual variances exceed that of S when $\mathrm{Var}_m(O) > 2|\,\mathrm{Cov}_m(S, O)|$. These identities explain a possible stability advantage, they do not by themselves establish an ordering of ROC curves or low-FPR detection. The identities compare the same per-coordinate loss estimates: since a one-sided score can use more draws per level under a fixed total query budget, variance diagnostics require a common number of draws per level, separately from the comparison at matched total query budgets.

Unequal reference distances. Consider references at offsets $-a$ and b, with $a, b > 0$. The equal-weight contrast $U_{x,a,b} = \ell_x(0) - [\ell_x(-a) + \ell_x(b)]/2$ retains $\beta_x(a - b)/2$ from an affine sample-dependent component $\alpha_x + \beta_x u$. Interpolation at the query point instead gives

$$T_{x,a,b} = \ell_x(0) - \frac{b}{a+b}\ell_x(-a) - \frac{a}{a+b}\ell_x(b), \tag{12}$$

which cancels every such affine component exactly. Applying Eq. equation 7 gives the finite-window identity

$$\begin{aligned} T_{x,a,b} &= \ell_x(0) - \frac{bE_x(a) + aE_x(b)}{a+b} + R_{x,a,b}, \\ R_{x,a,b} &= \frac{bO_x(a) - aO_x(b)}{a+b} = \frac{ab}{a+b}\left[\frac{O_x(a)}{a} - \frac{O_x(b)}{b}\right] \end{aligned} \tag{13}$$

Weighting therefore removes a linear odd component but generally retains nonlinear odd variation. For example, $O_x(u) = \beta_x u + c_x u^3$ contributes $R_{x,a,b} = c_x ab(a - b)$, whereas $a = b$ cancels every odd component. Unequal distances also sample the even component at different radii, which can change the retained membership contrast. Conversely, for an exactly quadratic profile $\ell_x(u) = A_x + B_x u + C_x u^2$, $T_{x,a,b} = -abC_x$ and $S_x = -\Delta^2 C_x$, so these two scores have identical rankings for fixed positive a, b, Δ.

Scope of the explanation. The benefit of symmetry depends on the removed odd variation being primarily nuisance and on the centered contrast retaining membership information. An odd membership component would also be removed, and sample-dependent even variation remains. Symmetry is therefore a useful invariance, not a universal optimality claim or a guarantee that all image difficulty is eliminated. In particular, smoothness of difficulty alone does not ensure its cancellation over a wide window. This account is compatible with a broad membership feature and does not identify a unique latent difficulty factor from observed losses.

These statements refer to symmetry in the evaluated log-SNR coordinates. Mapping requested levels to a discrete diffusion schedule can yield slightly unequal realized distances. In that case, a and b in Eq. equation 12 should be computed from the realized coordinate. Nominally equal spacing alone does not imply exact odd-component cancellation.

C.1 Symmetry Ablation Protocol

We use the same four held-out DDPM checkpoints at 4K epochs as the filter-selection ablation: CIFAR-10, CIFAR-100, STL-10-U, and Tiny ImageNet with horizontal flips. The query center is fixed at the deployed $\lambda^\star = 0.75$. The symmetric references have nominal distances $(a, b) = (2.5, 2.5)$. We compare each single reference separately and two predeclared unequal placements, $(a, b) = (1.5, 3.5)$ and $(3.5, 1.5)$, using either equal weights or the interpolation weights in Eq. equation 12. Thus all two-reference contrasts have the same nominal span $a + b = 5$. Neither the placement nor the score direction is selected using held-out membership labels, smaller residuals are always member-like.

At each noise level, we compute

$$\widehat{\ell}_{x,K}(\lambda) = \log\left(\frac{1}{K}\sum_{j=1}^{K} L_\theta(x, \lambda, \epsilon_j) + 10^{-12}\right). \tag{14}$$

At budgets $q \in \{15, 45\}$, three-point contrasts use $K = q/3$ draws at each level. One-reference contrasts use $\lceil q/2 \rceil$ draws at the center and $\lfloor q/2 \rfloor$ at the reference, assigning the remaining query to the center. Every configuration therefore uses exactly q model queries. The odd-component diagnostics instead use $q/3$ draws at all three levels, so the identities in Eq. equation 9 compare the same loss estimates.

Requested log-SNR levels are mapped to the nearest available DDPM timesteps. The nominal symmetric triple $(-1.75, 0.75, 3.25)$ is realized as approximately $(-1.752006, 0.755986, 3.264399)$. Its midpoint differs from the query level by approximately 2.11×10^{-4}. We retain equal weights for this deployed TSC baseline and use the realized distances for the unequal interpolation weights. We compute AUC and TPR@1%FPR separately for each checkpoint and then take their unweighted mean, rather than pooling scores across models.

Results by checkpoint. Table 11 reports the individual results underlying the main-text averages. Symmetry has the highest mean TPR@1%FPR across these four checkpoints at both budgets, but does not dominate every checkpoint or metric. The equal-weight $(1.5, 3.5)$ layout has higher AUC on all four checkpoints. The effect of interpolation weighting depends on the orientation and metric, consistent with its changing both nuisance variation and the membership contrast retained by the references.

Variation removed by the symmetric contrast. Table 12 compares the center log-loss, the symmetric contrast S, the endpoint difference O, and the two one-sided contrasts, using the same $K = 5$ draws at every point. For each checkpoint we compute the non-member minus member mean gap and the pooled within-class standard deviation $\sqrt{[\mathrm{Var}(S \mid M = 0) + \mathrm{Var}(S \mid M = 1)]/2}$, with analogous definitions for the other components, and then average over checkpoints. The left-only contrast retains a larger mean gap than S but has substantially greater spread. The smaller separation of O relative to its spread is consistent with suppression of difficulty-related variation, without establishing that O is pure nuisance. Its mean AUC is above chance, and covariance with S need not vanish. These are descriptive diagnostics at common per-point precision. The one-sided rows here use ten queries, whereas the symmetric contrast uses fifteen. They are distinct from the matched budget comparisons in Table 6.

Table 11: **Reference symmetry by checkpoint.** All checkpoints are held-out DDPMs at 4K epochs. AUC and TPR@1%FPR are percentages. C100 denotes CIFAR-100 with horizontal flips. Bold and underlined values denote the best and second-best results within each query budget.

		CIFAR-10		C100 hflip		STL-10-U		Tiny ImageNet	
q	Reference (a, b)	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR
15	Symmetric $(2.5, 2.5)$	<u>95.91</u>	**62.31**	95.85	**57.06**	<u>77.84</u>	**9.57**	<u>78.63</u>	<u>12.27</u>
15	Left only $(2.5, -)$	88.78	42.48	88.95	38.22	67.39	6.42	67.48	6.97
15	Right only $(-, 2.5)$	91.70	9.76	92.11	12.09	71.34	1.62	73.37	2.41
15	Unequal $(1.5, 3.5)$	**97.27**	<u>55.60</u>	**97.32**	<u>54.35</u>	**80.12**	6.79	**80.38**	7.02
15	Unequal $(3.5, 1.5)$	90.18	44.90	90.24	40.80	70.74	<u>8.63</u>	71.42	11.91
15	Weighted $(1.5, 3.5)$	95.71	52.60	<u>95.95</u>	48.96	77.26	7.67	76.04	7.21
15	Weighted $(3.5, 1.5)$	92.13	45.10	92.00	40.31	74.32	8.62	76.05	**12.31**
45	Symmetric $(2.5, 2.5)$	96.45	**67.80**	96.43	<u>62.60</u>	<u>79.60</u>	**12.40**	<u>79.96</u>	**14.26**
45	Left only $(2.5, -)$	89.35	44.31	89.47	40.82	68.05	6.61	68.04	7.53
45	Right only $(-, 2.5)$	91.99	10.29	92.51	11.96	71.76	1.64	73.78	2.32
45	Unequal $(1.5, 3.5)$	**97.96**	<u>65.65</u>	**97.93**	**62.88**	**81.16**	7.01	**82.04**	7.72
45	Unequal $(3.5, 1.5)$	91.03	48.08	91.03	45.22	71.72	10.05	71.90	13.02
45	Weighted $(1.5, 3.5)$	<u>96.89</u>	64.97	<u>97.06</u>	61.85	78.59	8.79	78.08	8.68
45	Weighted $(3.5, 1.5)$	93.39	51.12	93.06	46.58	76.03	<u>10.18</u>	76.73	<u>13.90</u>

Table 12: **Components of the temporal contrast.** Four-checkpoint means at $K = 5$ shared draws per point. Gaps and standard deviations are in log-loss units.

Component	Mean gap	Pooled SD	AUC
Center $\ell_x(0)$	0.3885	0.4119	74.26
Symmetric S	0.2566	0.1366	87.06
Endpoint difference O	0.0371	0.1552	57.14
Left reference $S + O$	0.2937	0.2464	77.80
Right reference $S - O$	0.2195	0.1580	81.84

D Prediction-Loss Analysis

This section expands the analysis in Section 4.3. We use $p \in \{x, v, \epsilon\}$ for the native prediction head and $s \in \{x, v, \epsilon\}$ for the training-loss target. The symbol $\mathcal{L}$ denotes squared error, while the calibration method uses ℓ for log-loss. q remains the number of model queries per candidate.

D.1 Conversions and Objective Weighting

For the linear flow $z_t = tx + (1-t)\epsilon$, with $v = x - \epsilon$ and $t \in (0, 1)$, the identities

$$x = z_t + (1-t)v, \qquad \epsilon = z_t - tv \tag{15}$$

give the conversions in Table 7. In particular, each native prediction induces a velocity prediction:

$$\widehat{v}_\theta^{(x)} = \frac{\widehat{x}_\theta - z_t}{1-t}, \qquad \widehat{v}_\theta^{(v)} = \widehat{v}_\theta, \qquad \widehat{v}_\theta^{(\epsilon)} = \frac{z_t - \widehat{\epsilon}_\theta}{t}. \tag{16}$$

Fix one model θ with native head p, and convert its output to each loss space. The resulting residuals satisfy

$$\delta_v^{(p)} = \widehat{v}_\theta^{(p)} - v, \qquad \delta_x^{(p)} = (1-t)\delta_v^{(p)}, \qquad \delta_\epsilon^{(p)} = -t\delta_v^{(p)}. \tag{17}$$

All three residuals here come from the same model output. This identity makes no assertion that separately trained prediction heads produce the same errors. For example, the expanded x-loss is

$$\begin{aligned} \mathcal{L}_{p,x}(x, \epsilon, t, \theta) &= \left\| \left(z_t + (1-t)\widehat{v}_\theta^{(p)}\right) - \left(z_t + (1-t)v\right) \right\|_2^2 \\ &= (1-t)^2 \left\| \delta_v^{(p)} \right\|_2^2. \end{aligned} \tag{18}$$

Squaring the other residuals gives the common-coordinate factorization used in Section 4.3:

$$\mathcal{L}_{p,s}(x,\epsilon,t,\theta) = w_s(t)\left\|\delta_v^{(p)}\right\|_2^2, \qquad (w_x(t), w_v(t), w_\epsilon(t)) = ((1-t)^2, 1, t^2). \tag{19}$$

For timestep sampling density $\rho(t)$, the population objective is therefore

$$\mathcal{R}_{p,s}(\theta) = \int_0^1 \rho(t) w_s(t) \mathbb{E}_{x,\epsilon}\left[\left\|\delta_v^{(p)}\right\|_2^2\right] dt. \tag{20}$$

The weight w_s is determined by the loss target, but the residual depends on the fitted model. Native prediction heads also have different output coordinates and parameter Jacobians. Thus, the common-coordinate factorization does not imply identical optimization or learned models. The analysis concerns interior timesteps, where the conversions are well-defined, before any numerical clamping at the endpoints.

For a fixed model and fixed interior timestep, evaluating its output in different loss spaces multiplies the squared error by a positive factor shared by all images. Averaging over the same noise draws preserves this relation, so the single-timestep rankings, ROC curves, and AUC are unchanged. This invariance concerns changing the evaluation space of one model. The nine configurations in Table 4 are separately trained models with different prediction/loss choices.

D.2 Temporal-Weight Barycenters and Their Ordering

For a common timestep sampling distribution ρ, define the barycenter of the explicit temporal weights as

$$c_s = \frac{\mathbb{E}_{T\sim\rho}[T w_s(T)]}{\mathbb{E}_{T\sim\rho}[w_s(T)]}. \tag{21}$$

This quantity summarizes the explicit weights $\rho(t)w_s(t)$, independently of the model-dependent residual in Eq. (20). For a sampling density $\rho(t)$, the corresponding normalized density is

$$\pi_s(t) = \frac{\rho(t)w_s(t)}{\int_0^1 \rho(u)w_s(u)\,du}, \qquad c_s = \int_0^1 t\pi_s(t)\,dt. \tag{22}$$

Uniform timestep sampling. For $\rho(t) = 1$, the normalization constants are $\int_0^1 (1-t)^2 dt = \int_0^1 t^2 dt = 1/3$, giving $\pi_x(t) = 3(1-t)^2$, $\pi_v(t) = 1$, and $\pi_\epsilon(t) = 3t^2$. Their barycenters are

$$\begin{aligned} c_x &= \frac{\int_0^1 t(1-t)^2\,dt}{\int_0^1 (1-t)^2\,dt} = \frac{1/12}{1/3} = \frac{1}{4}, \\ c_v &= \int_0^1 t\,dt = \frac{1}{2}, \\ c_\epsilon &= \frac{\int_0^1 t^3\,dt}{\int_0^1 t^2\,dt} = \frac{1/4}{1/3} = \frac{3}{4}. \end{aligned} \tag{23}$$

Thus the centers of the explicit weights move from the noisy toward the clean end as the training loss changes from x to v to ϵ. The factor t in these integrals measures the location of each weight, it is not an additional factor in the training objective.

General timestep sampling. Let $T \sim \rho$ have nonzero variance and take values in $(0,1)$. Since $w_v = 1$, $c_v = \mathbb{E}[T]$, and

$$c_s - c_v = \frac{\mathrm{Cov}(T, w_s(T))}{\mathbb{E}[w_s(T)]}. \tag{24}$$

For an independent copy T' of T and any function g used below,

$$2\,\mathrm{Cov}(T, g(T)) = \mathbb{E}\big[(T-T')\big(g(T) - g(T')\big)\big]. \tag{25}$$

For $g(t) = (1-t)^2$, which is strictly decreasing on $(0,1)$, the integrand is negative whenever $T \neq T'$. Conversely, for $g(t) = t^2$, which is strictly increasing, the integrand is positive. Hence

$$c_x < c_v < c_\epsilon. \tag{26}$$

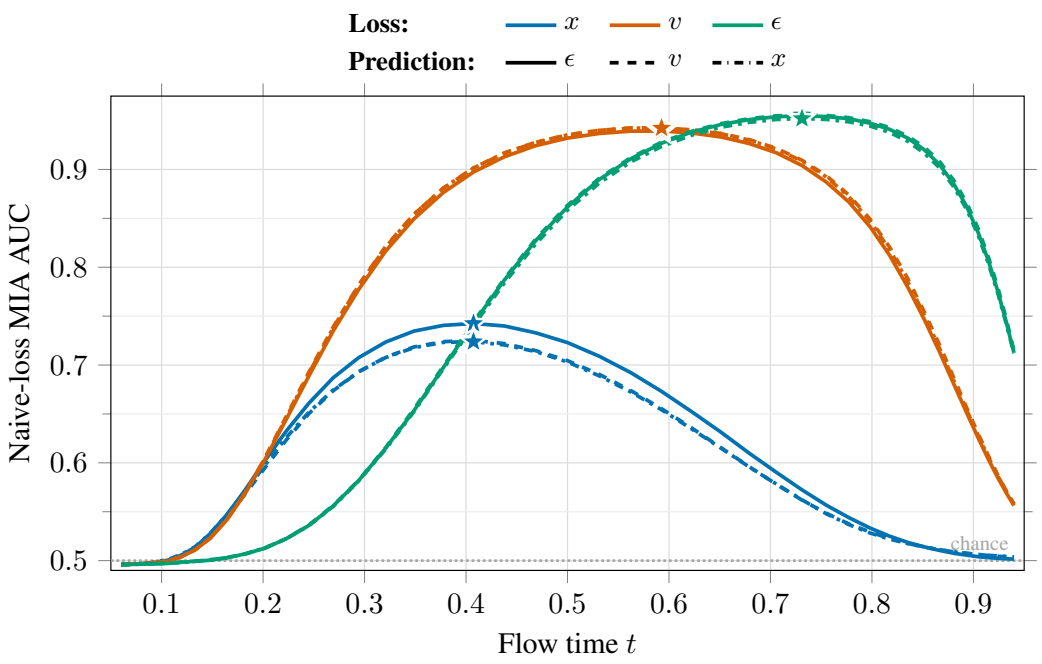

Figure 6: **Membership localization across training losses.** Single-timestep absolute-loss AUC on the nine first-half CIFAR-100 DDPM++ models without augmentation at epoch 2000. Color denotes training-loss target s and line style denotes native prediction p. Losses are averaged over 50 noise draws per image, shared across noise levels. Lower loss indicates membership at every level. Stars mark single-timestep AUC maxima. The peaks occur at $\lambda = -0.75$, 0.75, and 2.00 for x-, v-, and ϵ-loss, respectively.

For discrete sampling, replacing the integrals by sums gives the same proof. A distribution concentrated at one interior timestep instead gives equality throughout.

These centers describe only the explicit objective weighting. They are neither the centers of the actual training-error contribution nor the maxima of a membership-inference curve. The latter also depend on the learned errors, optimization, and sample information at each noise level. The ordering above is therefore a statement about explicit objective weights, not a theorem about membership peaks. It provides an interpretation of the empirical shifts examined below.

D.3 Empirical Localization of Membership Leakage

Figure 6 uses the nine CIFAR-100 DDPM++ models trained without augmentation for 2000 epochs. For each model, we evaluate the single-timestep absolute reconstruction loss in its training loss space on the grid $\mathcal{G} = \{-5.5, -5.25, \ldots, 5.5\}$ of 45 log-SNR values. Each curve uses 25,000 members and 25,000 non-members, with $K = 50$ deterministic Gaussian noise draws per image reused across noise levels. Losses are averaged over draws before taking the logarithm, this monotone transformation leaves single-timestep rankings unchanged.

Writing $\theta_{p,s}$ for the model trained with head p and loss target s, the underlying lower-loss membership score is

$$S_{p,s}(x,\lambda) = -\frac{1}{K}\sum_{k=1}^{K}\mathcal{L}_{p,s}\big(x,\epsilon_k,t(\lambda),\theta_{p,s}\big), \qquad t(\lambda) = \frac{1}{1+e^{-\lambda/2}}. \tag{27}$$

Let $A_{p,s}(\lambda)$ denote the ROC AUC of $S_{p,s}$, with members as the positive class. This fixes lower reconstruction loss as member-like at every noise level. The empirical leakage peak is

$$\lambda^{\mathrm{peak}}_{p,s} \in \arg\max_{\lambda\in\mathcal{G}} A_{p,s}(\lambda). \tag{28}$$

All three native heads attain the same discrete peak within each training loss: $\lambda^{\text{peak}} = -0.75$, 0.75, and 2.00 for x-, v-, and ϵ-loss, respectively. These correspond to flow times 0.407, 0.593, and 0.731. In this experiment, the membership signal therefore shifts toward higher log-SNR as the training loss changes from x to v to ϵ, consistent with the explicit-weight ordering. The observed peaks remain interior and differ from the barycenters. Their locations are empirical properties of the trained models and evaluation.

The empirical peak $\lambda_{p,s}^{\text{peak}}$ is measured separately for each model using the single-timestep score, it is distinct from the shadow-selected TSC center, which optimizes a three-point contrast. Appendix E.4 reports the full attack results across all nine prediction/loss configurations, with and without augmentation.

E Additional Experiments

Unless stated otherwise, all results in this section use the held-out second evaluation split, with operating points selected on the corresponding first-half shadow checkpoints. TSC uses the endpoint stencil with $\Delta = 2.5$ and five noise draws per trajectory point ($q = 15$). We do not apply frequency filtering to any attack. AUC, ASR, and TPR@1%FPR are reported as percentages. ASR uses the oracle threshold on the target evaluation split.

E.1 Earlier DDPM Checkpoints

Tables 13 and 14 complement the 4K-checkpoint results in the main paper with the corresponding 2K checkpoints. The shorter training horizon weakens membership leakage for every attack, particularly on STL-10-U and Tiny ImageNet, yet TSC remains the strongest method under every metric and dataset.

Table 13: **Membership inference at 2K DDPM checkpoints: CIFAR datasets.** Standard training setting, parentheses give model queries per candidate.

		CIFAR-10			CIFAR-100		
Method		AUC	ASR	TPR	AUC	ASR	TPR
Naive	($q = 20$)	70.70	65.44	2.75	69.31	64.18	4.19
SecMI	($q = 12$)	74.15	68.14	3.61	72.74	66.89	4.00
PIA	($q = 2$)	<u>77.09</u>	<u>70.75</u>	6.07	<u>75.82</u>	<u>69.48</u>	6.55
PIAN	($q = 2$)	74.24	69.18	<u>8.08</u>	72.75	68.07	<u>7.74</u>
TSC	($q = 15$)	**85.34**	**78.12**	**20.61**	**84.97**	**77.72**	**18.24**

Table 14: **Membership inference at 2K DDPM checkpoints: STL-10-U and Tiny ImageNet.** Same protocol as Table 13.

		STL-10-U			Tiny ImageNet		
Method		AUC	ASR	TPR	AUC	ASR	TPR
Naive	($q = 20$)	56.52	54.82	1.50	55.85	54.30	1.57
SecMI	($q = 12$)	<u>59.28</u>	<u>56.83</u>	1.63	56.85	54.95	1.52
PIA	($q = 2$)	59.15	56.68	1.84	<u>57.51</u>	<u>55.60</u>	1.84
PIAN	($q = 2$)	57.30	55.81	<u>1.95</u>	56.54	55.15	<u>2.05</u>
TSC	($q = 15$)	**63.70**	**60.11**	**3.30**	**62.07**	**59.03**	**2.84**

E.2 DDPMs Without Horizontal Flips

We additionally evaluate STL-10-U and Tiny ImageNet without horizontal-flip augmentation at both 2K and 4K epochs. The 4K checkpoints reproduce the no-flip training regime commonly used in prior diffusion-MIA evaluations. Tables 15 and 16 show that the advantage of TSC persists at both training horizons and becomes especially pronounced at low FPR after longer training.

Table 15: **Membership inference without horizontal flips at 2K epochs.** Parentheses give model queries per candidate.

		STL-10-U			Tiny ImageNet		
Method		AUC	ASR	TPR	AUC	ASR	TPR
Naive	$(q = 20)$	78.72	72.20	4.88	77.89	71.17	5.69
SecMI	$(q = 12)$	82.71	75.66	6.85	80.91	73.90	6.46
PIA	$(q = 2)$	<u>86.68</u>	<u>79.39</u>	12.59	<u>86.11</u>	<u>78.93</u>	13.52
PIAN	$(q = 2)$	83.72	77.85	<u>15.38</u>	83.29	77.53	<u>20.01</u>
TSC	$(q = 15)$	**93.39**	**86.88**	**38.23**	**93.28**	**86.55**	**45.92**

Table 16: **Membership inference without horizontal flips at 4K epochs.** Same protocol as Table 15.

		STL-10-U			Tiny ImageNet		
Method		AUC	ASR	TPR	AUC	ASR	TPR
Naive	$(q = 20)$	87.63	80.38	10.04	88.02	80.47	13.71
SecMI	$(q = 12)$	90.86	83.94	13.22	90.60	83.38	14.56
PIA	$(q = 2)$	<u>93.85</u>	<u>87.27</u>	29.10	<u>94.96</u>	<u>88.63</u>	40.48
PIAN	$(q = 2)$	91.64	85.81	<u>33.99</u>	91.98	87.00	<u>51.89</u>
TSC	$(q = 15)$	**96.98**	**91.97**	**56.91**	**97.59**	**92.63**	**77.07**

E.3 Conditioning and Diffusion Schedule

Class conditioning. We first score the same class-conditional ϵ-prediction, ϵ-loss SongUNet checkpoint either with the unconditional condition or with the ground-truth class label. Each evaluation contains 25K members and 25K non-members. Access to the class condition strengthens every attack, while TSC remains best in both settings.

Table 17: **Effect of class conditioning on a conditional DDPM++.** The same checkpoint is evaluated with the condition dropped or with the ground-truth class. TPR denotes TPR@1%FPR.

		Condition dropped			Ground-truth class		
Method		AUC	ASR	TPR	AUC	ASR	TPR
Naive	$(q = 20)$	67.93	63.17	4.74	77.83	71.05	8.16
SecMI	$(q = 12)$	62.19	58.61	3.25	68.25	63.02	5.06
PIA	$(q = 2)$	<u>74.86</u>	<u>68.89</u>	8.53	<u>85.29</u>	<u>77.52</u>	15.20
PIAN	$(q = 2)$	70.46	66.97	<u>8.90</u>	82.13	76.76	<u>18.90</u>
TSC	$(q = 15)$	**85.06**	**78.27**	**17.72**	**92.60**	**85.86**	**24.92**

Noise schedule. Table 18 compares class-conditional CIFAR-100 DDPMs trained for 2K epochs with cosine and linear noise schedules. Although the absolute leakage differs substantially between schedules, TSC provides the strongest separation in both cases.

Table 18: **Effect of the DDPM noise schedule.** Class-conditional CIFAR-100 checkpoints are evaluated at 2K epochs. TPR denotes TPR@1%FPR.

		Cosine schedule			Linear schedule		
Method		AUC	ASR	TPR	AUC	ASR	TPR
Naive	($q = 20$)	89.52	82.45	14.60	81.02	73.90	7.80
SecMI	($q = 12$)	88.76	81.11	17.08	82.74	75.49	8.69
PIA	($q = 2$)	95.16	88.99	30.86	86.40	78.88	12.80
PIAN	($q = 2$)	93.09	87.61	33.03	84.41	78.43	15.53
TSC	($q = 15$)	**98.73**	**94.91**	**75.32**	**95.29**	**89.08**	**37.20**

E.4 Full Prediction–Loss Results and Operating-Point Selection

The following tables report AUC, ASR, and TPR@1%FPR for each training-loss and native-prediction pair, with and without augmentation. The non-augmented models are trained for 2000 epochs, while the augmented counterparts are trained for 4000 epochs.

Naive uses 20 noise draws at one level ($q = 20$). TSC uses five draws at each of its three levels ($q = 15$), with $\Delta = 2.5$ and no frequency filtering. PIA and PIAN use their two-query scores. SecMI follows its original fixed protocol ($q = 12$), so no center is selected for it. Table 4 summarizes the non-augmented TPR@1%FPR results. The following tables below retain all metrics and the augmented configurations. Each loss uses one consistent point λ^* shared across prediction parameterizations and augmentation variants. All metrics are in percentages and TPR denotes TPR@1%FPR. Augmentations are the default EDM (Karras et al., 2022) augmentations.

Table 19: **DDPM++ results under x-loss without augmentation.**

		$\lambda^\star$	x-pred			v-pred			ϵ-pred		
Method			AUC	ASR	TPR	AUC	ASR	TPR	AUC	ASR	TPR
Naive	($q = 20$)	-0.75	72.20	66.16	6.48	71.77	65.80	6.05	73.06	66.75	6.74
SecMI	($q = 12$)	fixed	50.57	50.67	0.94	51.78	51.38	1.10	51.03	50.98	1.19
PIA	($q = 2$)	-0.75	58.52	57.03	2.27	72.84	67.11	4.07	74.00	67.87	6.22
PIAN	($q = 2$)	-0.5	55.88	55.17	2.41	68.38	64.15	6.00	67.49	63.63	5.69
TSC	($q = 15$)	-0.5	**81.96**	**73.94**	**30.28**	**81.34**	**73.50**	**28.74**	**82.60**	**74.52**	**31.26**

Table 20: **DDPM++ results under x-loss with augmentation.**

		$\lambda^\star$	x-pred			v-pred			ϵ-pred		
Method			AUC	ASR	TPR	AUC	ASR	TPR	AUC	ASR	TPR
Naive	($q = 20$)	-0.75	56.70	54.85	1.96	56.85	54.95	1.85	57.47	55.48	1.98
SecMI	($q = 12$)	fixed	50.37	50.40	1.01	50.58	50.63	1.12	50.58	50.64	1.02
PIA	($q = 2$)	-0.75	52.25	51.77	1.14	57.57	55.64	1.75	58.06	56.01	1.94
PIAN	($q = 2$)	-0.5	50.92	50.82	1.07	54.91	53.80	1.54	56.39	54.76	1.64
TSC	($q = 15$)	-0.5	**61.47**	**58.32**	**5.71**	**61.58**	**58.30**	**5.82**	**62.43**	**59.11**	**6.36**

Table 21: **DDPM++ results under v-loss without augmentation.**

Method		$\lambda^\star$	x-pred			v-pred			ϵ-pred		
			AUC	ASR	TPR	AUC	ASR	TPR	AUC	ASR	TPR
Naive	($q = 20$)	0.25	93.75	86.19	34.54	93.77	86.25	34.28	93.57	86.02	34.56
SecMI	($q = 12$)	fixed	69.75	63.99	5.55	74.93	68.28	5.57	73.66	67.23	5.26
PIA	($q = 2$)	0.25	50.23	50.37	1.01	95.73	89.26	52.48	96.30	90.05	54.84
PIAN	($q = 2$)	0.5	50.36	50.39	1.06	94.31	88.44	56.68	93.81	88.13	58.36
TSC	($q = 15$)	0.5	**98.04**	**93.03**	**81.31**	**97.89**	**92.83**	**80.56**	**97.88**	**92.71**	**80.68**

Table 22: **DDPM++ results under v-loss with augmentation.**

Method		$\lambda^\star$	x-pred			v-pred			ϵ-pred		
			AUC	ASR	TPR	AUC	ASR	TPR	AUC	ASR	TPR
Naive	($q = 20$)	0.25	69.69	64.32	5.97	69.77	64.35	5.88	70.04	64.59	5.96
SecMI	($q = 12$)	fixed	50.53	50.54	1.13	50.64	50.70	1.09	51.04	50.88	1.16
PIA	($q = 2$)	0.25	50.26	50.38	1.00	74.43	68.35	8.32	74.51	68.29	6.60
PIAN	($q = 2$)	0.5	50.25	50.38	1.13	69.94	65.59	8.18	69.19	65.24	8.31
TSC	($q = 15$)	0.5	**80.80**	**73.97**	**19.42**	**80.73**	**73.95**	**18.45**	**80.94**	**74.20**	**19.75**

Table 23: **DDPM++ results under ϵ-loss without augmentation.**

Method		$\lambda^\star$	x-pred			v-pred			ϵ-pred		
			AUC	ASR	TPR	AUC	ASR	TPR	AUC	ASR	TPR
Naive	($q = 20$)	1.5	94.67	88.42	27.52	95.16	89.15	29.78	94.93	88.77	28.29
SecMI	($q = 12$)	fixed	94.64	88.24	29.25	96.24	90.68	35.40	95.42	89.43	30.82
PIA	($q = 2$)	1.75	63.63	66.81	31.69	98.65	94.68	74.61	98.72	94.77	75.85
PIAN	($q = 2$)	1.75	93.08	90.53	72.11	95.29	92.41	77.77	94.60	91.88	76.73
TSC	($q = 15$)	1.75	**99.37**	**96.68**	**91.22**	**99.52**	**97.18**	**93.22**	**99.49**	**97.06**	**92.75**

Table 24: **DDPM++ results under ϵ-loss with augmentation.**

Method		$\lambda^\star$	x-pred			v-pred			ϵ-pred		
			AUC	ASR	TPR	AUC	ASR	TPR	AUC	ASR	TPR
Naive	($q = 20$)	1.5	71.95	66.28	6.39	72.58	66.68	6.36	72.99	67.00	6.91
SecMI	($q = 12$)	fixed	64.86	60.36	4.16	67.70	62.54	4.69	67.46	62.39	4.79
PIA	($q = 2$)	1.75	53.51	53.11	3.44	80.70	73.49	13.71	81.23	74.05	13.36
PIAN	($q = 2$)	1.75	72.38	70.19	13.50	75.17	71.33	14.36	74.14	70.92	15.54
TSC	($q = 15$)	1.75	**88.53**	**81.54**	**22.41**	**89.20**	**82.26**	**24.12**	**89.35**	**82.48**	**23.28**

E.5 Reference Calibration Across Noise Levels

We evaluate RMC across 45 noise levels for all three realistic Stable Diffusion fine-tunes in Table 1, using the same target checkpoints. The nominal log-SNR grid spans $[-5.5, 5.5]$ in increments of 0.25, with each coordinate mapped to the nearest available diffusion timestep. At each level, RMC uses one shared noise draw for the fine-tuned and pretrained models ($q = 2$), without captions. We also report the uncalibrated fine-tuned loss from the same draw ($q = 1$) to show the effect of subtracting the reference loss.

Around the operating region reported in Table 1, RMC substantially improves both metrics over the uncalibrated loss on all three datasets. The curves also characterize the sensitivity to the chosen noise level: reference calibration reduces difficulty-related variation, while the available membership signal can still vary along the diffusion trajectory.

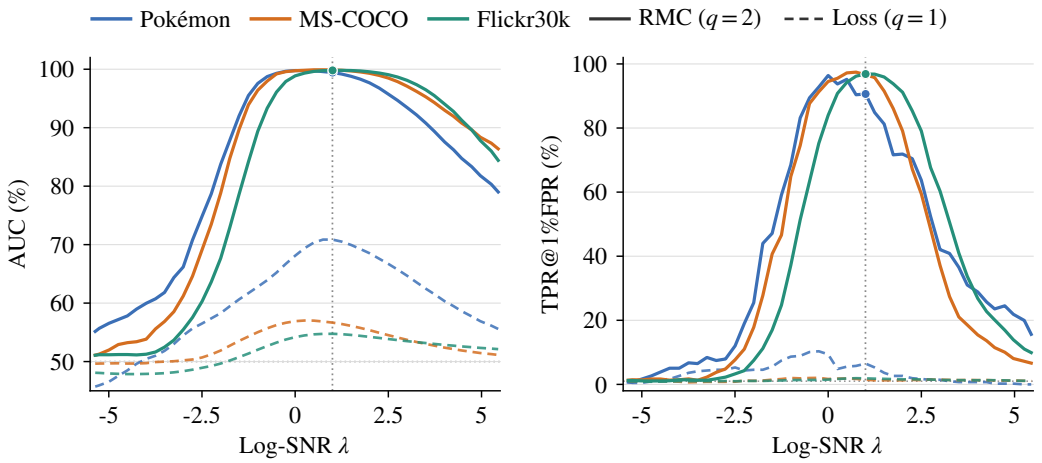

Figure 7: **Reference calibration across noise levels.** AUC (left) and TPR@1%FPR (right) for the realistic Pokémon, MS-COCO, and Flickr30k fine-tunes. Solid curves show RMC ($q = 2$). Dashed curves show the uncalibrated fine-tuned loss from the same noise draw ($q = 1$). Each point uses a fixed score direction, with lower loss or calibrated residual indicating membership. The vertical guide marks $\lambda = 1$, the operating point reported in Table 1.