Characterizing Memorization in Diffusion Language Models: Generalized Extraction and Sampling Effects

Xiaoyu Luo[1,*], **Wenrui Yu**[2,*], **Qiongxiu Li**[2,†], **Johannes Bjerva**[1]
[1]Department of Computer Science, [2]Department of Electronic Systems
Aalborg University, Copenhagen, Denmark
{xilu,jbjerva}@cs.aau.dk, {wenyu,qili}@es.aau.dk

Abstract

Autoregressive language models (ARMs) have been shown to memorize and occasionally reproduce training data verbatim, raising concerns about privacy and copyright liability. Diffusion language models (DLMs) have recently emerged as a competitive alternative, yet their memorization behavior remains largely unexplored due to fundamental differences in generation dynamics. To address this gap, we present a systematic theoretical and empirical characterization of memorization in DLMs. We propose a generalized probabilistic extraction framework that unifies prefix-conditioned decoding and diffusion-based generation under arbitrary masking patterns and stochastic sampling trajectories. Theorem 4.3 establishes a monotonic relationship between sampling resolution and memorization: increasing resolution increases the probability of exact training data extraction, implying that autoregressive decoding corresponds to a limiting case of diffusion-based generation by setting the sampling resolution maximal. Extensive experiments across model scales and sampling strategies validate our theoretical predictions. Under aligned prefix-conditioned evaluations, we further demonstrate that DLMs exhibit substantially lower memorization-based leakage of personally identifiable information (PII) compared to ARMs.

1 Introduction

Large Language Models (LLMs) have achieved remarkable success and have been integrated into a wide range of everyday applications. Despite this progress, the field has long been dominated by the decoder-only architecture under the ARMs paradigm Radford et al. [2018, 2019], Brown et al. [2020]. More recently, diffusion principles Ho et al. [2020] have been successfully incorporated into masked DLMs Lou et al. [2023], Shi et al. [2024], Ou et al. [2024], leading to the emergence of Large DLMs Nie et al. [2025a,b]. DLMs have demonstrated competitive scalability and performance with fewer training tokens. Since the training paradigm of these models fundamentally shifts from unidirectional next-token prediction to a bidirectional masking and reverse denoising process, their memorization behavior remains a critical yet unexplored frontier. Unlike ARMs that minimize the negative log-likelihood of sequential conditionals, DLMs optimizes a variational lower bound through a non-causal masking objective Ou et al. [2024]. This fundamental difference in how models internalize data necessitates a formal definition and systematic investigation of their memorization behavior, as it remains unclear how bidirectional denoising affects exposure to training data.

In the context of ARMs, a substantial body of work has shown that models can memorize and reproduce training samples verbatim Carlini et al. [2021, 2022], Kiyomaru et al. [2024], Huang et al. [2024], Li et al. [2024], Zhang et al. [2025a], Luo et al. [2025]. Accordingly, studying memorization

[*]Equal contribution. [†]Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

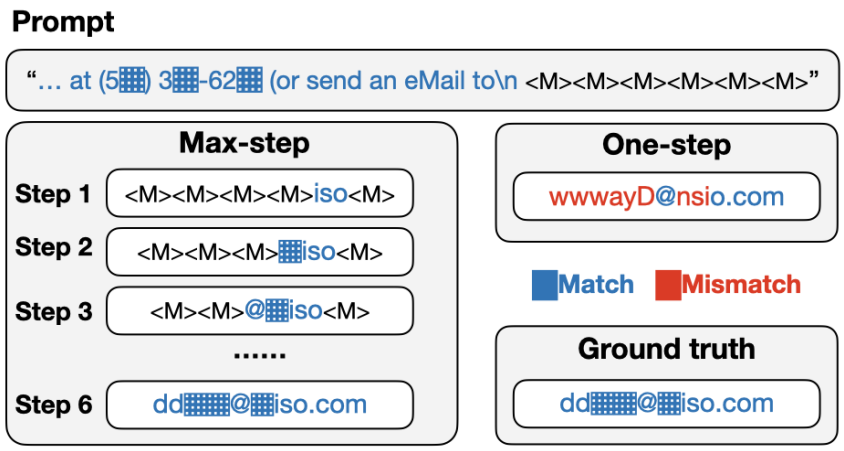


Figure 1: A randomly sampled verbatim PII memorization example from diffusion language model (LLaDA-8B). This case shows that finer-grained generation resolution (i.e., more steps) is more prone to verbatim recovery of training data. ⟨M⟩ denotes a mask token; blue highlights matched strings, and red highlights mismatches.

is a practical necessity, as it is closely tied to privacy leakage Huang et al. [2022], Kim et al. [2023] and copyright infringement Cooper et al. [2025], Ahmed et al. [2026].

Standard discoverable extraction procedures typically rely on prefix–suffix prompting Carlini et al. [2022], Hayes et al. [2025]. In contrast, DLMs differ in both conditioning and sampling, so we introduce an adapted evaluation framework to measure discoverable extraction under their bidirectional denoising paradigm.

In this work, we present the first systematic investigation into the memorization behavior of diffusion language models (DLMs). Our contributions are three-fold:

- **A generalized formulation of memorization for diffusion language models.** We develop a generalized probabilistic formulation of discoverable memorization that applies to DLMs with arbitrary masking patterns and stochastic sampling trajectories, under which the prefix-conditioned (n, p)-discoverable extraction for ARMs arises as a special case.
- **Sampling resolution provably controls memorization.** We prove a monotonicity result showing that, under a natural recovery observation, increasing the sampling resolution (i.e., recovering masked tokens in finer-grained denoising steps) generally increases the probability of exact memorization, and we demonstrate this effect empirically across sampling strategies and model scales.
- **Aligned analysis of PII memorization across model scales and architectures.** We align autoregressive and diffusion language models under the same prefix-conditioned PII completion setting and quantify memorization across multiple model scales and architectures, finding that autoregressive models exhibit higher PII memorization and leakage risk.

2 Related Work

Diffusion language models. Recent work has scaled diffusion-based text generation toward the regime of large language models. Nie et al. [2025a] establish scaling laws for masked diffusion language models, showing ARM-like power-law improvement with model and data scale. Building on this direction, LLaDA [Nie et al., 2025b] adopts a modern LLM training recipe with large-scale pretraining followed by supervised fine-tuning, scales DLMs to 8B parameters, and reports competitive zero-/few-shot and in-context learning performance against same-scale ARM baselines and 8B-class models such as LLaMA3 [AI@Meta, 2024]. Its bidirectional conditioning also improves robustness in directionality-sensitive settings, including reversal-style completion tasks relative to GPT-4o [OpenAI, 2024]. More recently, LLaDA2.0 [Bie et al., 2025] further scales DLMs to 16B/100B and shows continued performance gains while retaining the parallel-generation advantages

of diffusion decoding. These advances make it increasingly important to understand whether DLMs exhibit memorization risks comparable to those observed in autoregressive models.

Memorization in language models. Prior work has shown that language models can memorize and reproduce portions of their training data, including rare, sensitive, and copyrighted content [Carlini et al., 2021, 2022, Huang et al., 2024, Li et al., 2024, Zhang et al., 2025a]. Early extraction attacks demonstrate that black-box access can recover verbatim training sequences from generative models [Carlini et al., 2021], while subsequent work quantifies how extractable memorization increases with model scale and data repetition [Carlini et al., 2022]. Beyond controlled verbatim extraction, another line of work studies personally identifiable information (PII) leakage, showing that private attributes may be recovered through association-based or indirect prompting attacks [Huang et al., 2022, Kim et al., 2023, Lukas et al., 2023]. At the same time, prompt cues can inflate apparent memorization estimates, motivating stricter cue control when measuring privacy leakage [Luo et al., 2026].

Memorization has also been studied in deployed and aligned language models. Recent work shows that alignment safeguards and production interfaces do not fully prevent extraction of memorized training examples [Nasr et al., 2025], and analyses of open-weight or deployed systems find that copyrighted books can sometimes be reconstructed to a substantial extent under probabilistic extraction or adversarial prompting [Cooper et al., 2025, Ahmed et al., 2026]. Most of these evaluations, however, are designed around autoregressive decoding, often using greedy or single-query prefix-conditioned extraction. Hayes et al. [2025] address this limitation by introducing probabilistic extractable memorization, which measures recovery under stochastic decoding and repeated querying. Our work builds on this probabilistic view, but adapts it to diffusion language models: whereas ARMs induce a fixed left-to-right continuation path, DLM extraction depends on arbitrary masking patterns, stochastic denoising trajectories, and sampling resolution.

3 Preliminaries

In this section, we briefly review DLMs and ARMs architectures, introduce their commonly used sampling paradigms, and finally present the existing (n, p)-discoverable extraction framework defined for ARMs.

3.1 Masked Diffusion Model

We review the framework of the masked diffusion paradigm Ou et al. [2024], Nie et al. [2025b], which serves as the fundamental architecture for Large DLMs.

Forward Process. Let $\mathcal{V} = \{0, 1, \ldots, K-1\}$ be the vocabulary set. Given a sequence $\boldsymbol{z}_0 \in \mathcal{V}^L$ of length L, the forward process at noise level $t \in [0, 1]$ gradually replaces tokens. The transition probability is defined as

$$q_{t|0}(\boldsymbol{z}_t|\boldsymbol{z}_0) = \prod_{l=1}^{L} q_{t|0}(z_t^l|z_0^l), \tag{1}$$

where $q_{t|0}(z_t^l|z_0^l) = \begin{cases} \alpha_t, & z_t^l = z_0^l \\ 1-\alpha_t, & z_t^l = m \end{cases}$. $\boldsymbol{z}_t$ denotes the noisy data at time t, z_t^l denote the l-th element of $\boldsymbol{z}_t$, m denotes a special [MASK] token Devlin et al. [2019] and $\alpha_t = 1 - t$.

Reverse Process. The reverse process will recover the original tokens $\boldsymbol{z}_0$ from a fully masked sequence $\boldsymbol{z}_1$. For any $0 \le s < t \le 1$, the transition $q_{s|t}(\boldsymbol{z}_s|\boldsymbol{z}_t)$ recovers a portion of the masked tokens, which can be formulated as

$$q_{s|t}(z_s^l|\boldsymbol{z}_t) = \begin{cases} 1, & z_t^l \neq m, z_s^l = z_t^l \\ \frac{s}{t}, & z_t^l = m, z_s^l = m \\ \frac{t-s}{t} q_{0|t}(z_s^l|\boldsymbol{z}_t), & z_t^l = m, z_s^l \neq m \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Training Objective. A model $p_\theta(z_0^l|\boldsymbol{z}_t)$ is trained to approximate the true data distribution by minimizing the following negative log-likelihood upper bound as

$$\mathcal{L} = \int_0^1 \frac{1}{t} \mathbb{E}_{q(\boldsymbol{z}_t|\boldsymbol{z}_0)} \left[\sum_{l:z_t^l=m} -\log p_\theta(z_0^l|\boldsymbol{z}_t) \right] dt. \tag{3}$$

Sampling Strategy. Sampling iteratively updates the currently masked positions while keeping unmasked tokens unchanged. A *linearly spaced* time grid $\{t_i\}_{i=0}^N$ is used, with $t_0 = 1$ and $t_{i+1} < t_i$, N is the sampling resolution. The linear schedule determines the fraction of tokens to be recovered at each step $t \to s$ by $1 - \frac{s}{t}$. At step $t \to s$, for each masked position l ($z_t^l = m$), the model predicts a token $\hat{z}_0^l = \arg\max_{v \in \mathcal{V}} p_\theta(v \mid \boldsymbol{z}_t)$ with confidence $p_\theta(\hat{z}_0^l \mid \boldsymbol{z}_t)$. *Greedy sampling* selects the masked positions with the highest confidence and recovers $k = \lfloor |\mathcal{M}|(1 - \frac{s}{t}) \rfloor$ tokens, while *random sampling* recovers the same number of tokens by uniformly sampling positions from $\mathcal{M}$.

3.2 Autoregressive Model

Most existing LLMs follow the ARM formulation Radford et al. [2018]. Given a sequence $\boldsymbol{z} \in \mathcal{V}^L$, an ARMs factorizes

$$p_\theta(\boldsymbol{z}) = \prod_{l=1}^{L} p_\theta\left(z^l \mid \boldsymbol{z}^{1:l-1}\right), \tag{4}$$

where $\boldsymbol{z}^{1:l-1} = [z^1, \ldots, z^{l-1}]$ is the prefix; we use this as the canonical baseline paradigm throughout.

3.3 Sampling Paradigm

We consider both deterministic and stochastic decoding. For ARMs, stochasticity is induced by standard temperature or top-k sampling; for DLMs, it is induced by random mask selection and Gumbel-perturbed token sampling. Full decoding details are given in Appendix B.

3.4 (n, p)-discoverable Extraction Framework

We review two definitions of memorization for ARMs as introduced by Hayes et al. [2025].

Let $[L] = \{1, 2, \ldots, L\}$ be the set of all token indices and $\boldsymbol{z} = [z_1, \ldots, z_L] \in \mathcal{V}^L$ be a training example.

Definition 3.1 ((n, p)-discoverable Extraction)**.** A sequence is (n, p)-discoverably extractable if, given a prefix, the model generates the exact suffix within n independent queries with probability at least p.

Let $p_{\boldsymbol{z}}$ denote the probability of generating the exact suffix in a single trial. Consequently, the probability of failing to recover the correct suffix in n independent trials is $(1-p_{\boldsymbol{z}})^n$. In order to ensure that the probability of at least one successful extraction is at least p, we require $1 - (1 - p_{\boldsymbol{z}})^n \geq p$. Solving for n, we obtain the estimation for the required number of queries as

$$n \geq \frac{\log(1-p)}{\log(1-p_{\boldsymbol{z}})}. \tag{5}$$

This relationship implies that the extractability parameters (n, p) can be determined once $p_{\boldsymbol{z}}$ is known. In practice, $p_{\boldsymbol{z}}$ is computed as the cumulative product of the conditional probabilities associated with recovering each token at its corresponding sampling step.

Beside the exact extraction, Hayes et al. [2025] also proposed a relaxed version for approximate recovery.

Definition 3.2 ((ϵ, n, p)-discoverable Extraction)**.** A suffix of an example is (ϵ, n, p)-discoverably extractable if the probability that at least one of n independent queries results in a generated suffix with a distance $\leq \epsilon$ from the ground truth is at least p.

4 A Generalized Discoverable Memorization Framework for Diffusion Language Models

Existing definitions of discoverable memorization [Hayes et al., 2025] are inherently tied to the autoregressive paradigm, where memorization is evaluated through a fixed prefix–suffix structure under left-to-right decoding. While this formulation closely mirrors the unidirectional generation process of autoregressive language models, it does not naturally extend to diffusion language models, which recover masked tokens through stochastic, non-sequential denoising trajectories.

To address this mismatch, we develop a generalized formulation of discoverable memorization that removes the prefix-based decoding assumption and explicitly accounts for arbitrary masking patterns and diffusion sampling trajectories. Moreover, within our proposed framework, prefix-conditioned extraction can be recovered as a special case (see Remark 4.4), allowing memorization in diffusion language models to be analyzed under their native generation dynamics.

4.1 Generalized Exact Memorization in DLMs

We first focus on the case of exact memorization.

4.1.1 Definition and Approximation

Definition 4.1 (Generalized (n, p)-discoverable Extraction)**.** A sequence $\boldsymbol{z}$ is (n, p)-discoverably extractable under a mask $\mathcal{M} \subset \{1, \ldots, L\}$ if, given the observed tokens $\boldsymbol{z}_{\bar{\mathcal{M}}}$, the model recovers the exact original tokens at all masked positions $\mathcal{M}$ within n independent queries with probability at least p.

Formally, let $\hat{\boldsymbol{z}}^{(i)}_{\mathcal{M}}$ be the output of the i-th independent sampling. The sequence $\boldsymbol{z}$ is (n, p)-extractable if:

$$\Pr\left(\exists i \in \{1, \ldots, n\} : \hat{\boldsymbol{z}}^{(i)}_{\mathcal{M}} = \boldsymbol{z}_{\mathcal{M}} \mid \boldsymbol{z}_{\bar{\mathcal{M}}}\right) \geq p, \tag{6}$$

where $\bar{\mathcal{M}} = [L] \backslash \mathcal{M}$ denotes the complement set of indices (the observed positions).

Following Hayes et al. [2025], the required number of independent queries n can be estimated given a target extraction probability p and an approximation of the single-trial recovery probability $p_{\boldsymbol{z}}$. In the context of diffusion-based LLMs, the approximation of $p_{\boldsymbol{z}}$ is a bit different and it is inherently dependent on the sampling resolution N.

For a sampling process with N steps in the i-th independent query, we partition $\mathcal{M}$ into N disjoint chunks $\{\Delta^{(i)}_1, \Delta^{(i)}_2, \ldots, \Delta^{(i)}_N\}$ such that $\Delta^{(i)}_k$ represents the set of token indices recovered at step k. Let $\mathcal{U}^{(i)}_k$ be the set of observed indices after step k, with $\mathcal{U}^{(i)}_0 = [L] \backslash \mathcal{M}$ and $\mathcal{U}^{(i)}_k = \mathcal{U}^{(i)}_{k-1} \cup \Delta^{(i)}_k$. Because tokens are recovered independently at each step (given the unmasked context), the total probability $p^{(i)}_{\boldsymbol{z}}$ is defined by the multiplicative product of their respective conditional probabilities

$$\begin{aligned} p^{(i)}_{\boldsymbol{z}} &= \Pr(\hat{\boldsymbol{z}}_{\mathcal{M}} = \boldsymbol{z}_{\mathcal{M}} \mid \boldsymbol{z}_{\bar{\mathcal{M}}}, N) \\ &= \prod_{k=1}^{N} \prod_{\pi \in \Delta^{(i)}_k} \Pr(\hat{z}_\pi = z_\pi \mid \boldsymbol{z}_{\mathcal{U}^{(i)}_{k-1}}). \end{aligned} \tag{7}$$

Unlike ARMs, where the decoding order is typically fixed, diffusion-based models exhibit stochasticity in the recovery path (i.e., the sequence of token subsets selected at each step). Consequently, the single-trial probability $p_{\boldsymbol{z}}$ cannot be reliably estimated from a single query, as is possible in the deterministic setting of autoregressive. To account for this randomness, we estimate the expected $\hat{p}_{\boldsymbol{z}}$ by averaging results over a set of R trials, i.e. $\hat{p}_{\boldsymbol{z}} = \frac{1}{R} \sum_{i=1}^{R} p^{(i)}_{\boldsymbol{z}}$. The validation results are in Sec 6.1.

4.1.2 The Impact of Sampling Resolution

Remark 4.2 (Monotonicity of Recovery Probability). We assume that, for a given inference step, additional correct context does not decrease the probability of correctly recovering the remaining masked tokens.

Formally, for any two observation index sets $\mathcal{U}, \mathcal{W} \subseteq [L]$ such that $\mathcal{U} \subset \mathcal{W}$, and for any target mask set $\mathcal{M} \subseteq [L] \setminus \mathcal{W}$, we have

$$\Pr(\hat{\boldsymbol{z}}_{\mathcal{M}} = \boldsymbol{z}_{\mathcal{M}} \mid \boldsymbol{z}_{\mathcal{W}}) \geq \Pr(\hat{\boldsymbol{z}}_{\mathcal{M}} = \boldsymbol{z}_{\mathcal{M}} \mid \boldsymbol{z}_{\mathcal{U}}) \tag{8}$$

where $\boldsymbol{z}_{\mathcal{U}}$ and $\boldsymbol{z}_{\mathcal{W}}$ denotes the sub-vector of tokens at indices in $\mathcal{U}$ and $\mathcal{W}$ respectively, and $\hat{\boldsymbol{z}}_{\mathcal{M}}$ is the reconstruction of tokens at indices $\mathcal{M}$.

This remark formalizes the intuitive property that additional *correct context* should not reduce recovery likelihood, which is also validated empirically in Sec 6.2. It is important to note that we specifically consider context consisting of correctly recovered tokens, rather than potentially erroneous predictions. This focus is consistent with our objective of analyzing *exact memorization*, where the recovery of the entire sequence is only successful if every intermediate step along the sampling trajectory remains faithful to the training instance.

Hereby we have the following Theorem.

Theorem 4.3 (Impact of Sampling Resolution on Memorization)**.** *Under the monotonicity property in Remark 4.2, for diffusion-based LLMs, the probability of generating a sequence that exactly matches a training instance generally increases with the number of sampling steps N.*

In certain cases, this relationship is theoretically monotonic under a fixed recovery sequence. Let $\sigma = (\pi_1, \pi_2, \ldots, \pi_M)$ be a fixed permutation of indices in $\mathcal{M}$ representing the recovery order. For any N, we partition σ into N disjoint and ordered chunks $\mathcal{P}_N = \{\Delta_1, \Delta_2, \ldots, \Delta_N\}$. If $N_1 > N_2$ and the partition $\mathcal{P}_{N_1}$ is a refinement of $\mathcal{P}_{N_2}$ (i.e., each chunk in $\mathcal{P}_{N_2}$ is the union of one or more consecutive chunks in $\mathcal{P}_{N_1}$), then:

$$Pr(\hat{\boldsymbol{z}}_{\mathcal{M}} = \boldsymbol{z}_{\mathcal{M}} \mid \boldsymbol{z}_{\bar{\mathcal{M}}}, N_1) \geq Pr(\hat{\boldsymbol{z}}_{\mathcal{M}} = \boldsymbol{z}_{\mathcal{M}} \mid \boldsymbol{z}_{\bar{\mathcal{M}}}, N_2). \tag{9}$$

Proof. See Appendix A. □

This theorem establishes that increasing the sampling resolution consistently enhances the probability of exact memorization. Since the number of sampling steps N is upper-bounded by the total number of masked tokens $|\mathcal{M}|$, the most granular refinement occurs when $N = |\mathcal{M}|$. In this limiting case, the diffusion model recovers tokens in one-by-one fashion, mirroring the fundamental mechanism of ARMs. This leads to the following remark.

Remark 4.4. At the maximum sampling resolution $N = |M|$, a DLM recovers one masked token at each step. Unlike autoregressive decoding, the recovery order is not necessarily left-to-right and may follow an arbitrary permutation. When the masked region corresponds to a suffix and the recovery order is fixed to be left-to-right, this sequential recovery schedule corresponds to the autoregressive decoding paradigm. Therefore, ARM-style decoding can be viewed as a special case of the maximum-resolution DLM sampling framework.

4.2 Generalized Relaxed Memorization in DLMs

Definition 4.5 (Generalized (ϵ, n, p)-discoverable Extraction)**.** We generalize the notion of discoverable extraction to arbitrary masking patterns. An example $\boldsymbol{z}$ is (ϵ, n, p)-discoverably extractable under a mask $\mathcal{M}$ if, given the observed tokens $\boldsymbol{z}_{\bar{\mathcal{M}}}$, the model generates a reconstructed sequence $\hat{\boldsymbol{z}}_{\mathcal{M}}$ such that the distance to the original tokens $d(\hat{\boldsymbol{z}}_{\mathcal{M}}, \boldsymbol{z}_{\mathcal{M}}) \leq \epsilon$ holds for at least one of n independent queries with probability at least p, i.e.,

$$\Pr\left(\exists i \in \{1, \ldots, n\} : d(\hat{\boldsymbol{z}}^{(i)}_{\mathcal{M}}, \boldsymbol{z}_{\mathcal{M}}) \leq \epsilon \mid \boldsymbol{z}_{\bar{\mathcal{M}}}\right) \geq p, \tag{10}$$

where $d(\cdot, \cdot)$ is a distance metric. When $\epsilon = 0$, this definition reduces to the case of exact extraction.

In Hayes et al. [2025], the error-tolerant extraction probability $p_{\boldsymbol{z},\epsilon}$ can be empirically estimated by calculating the proportion of successful reconstructions (the distance is within the threshold ϵ) across R independent trials, i.e.,

$$\hat{p}_{\boldsymbol{z},\epsilon} = \frac{1}{R} \sum_{i=1}^{R} \mathbb{I}\left(d(\hat{\boldsymbol{z}}^{(i)}_{\mathcal{M}}, \boldsymbol{z}_{\mathcal{M}}) \leq \epsilon\right). \tag{11}$$

This estimation method naturally extends to the generalized definition. Specifically, when the error tolerance is set to $\epsilon = 0$, the estimator $\hat{p}_{\boldsymbol{z},0}$ provides a direct empirical approximation for the exact memorization probability $\Pr(\hat{\boldsymbol{z}}_{\mathcal{M}} = \boldsymbol{z}_{\mathcal{M}} \mid \boldsymbol{z}_{\bar{\mathcal{M}}}, N)$ with a large R.

In addition, we investigate the statistical behavior of the error distribution under large-scale trials. We refer the reader to Appendix C for the related experimental analysis.

5 Experimental Setup

Foundation model (DLM and ARM). We pretrain DLMs with 170M, 629M, and 1.1B parameters on SlimPajama following the recipe of Nie et al. [2025a], under a fixed compute budget of 10^{21} FLOPs. For the controlled architecture comparison, we train a 1.1B ARM with the same dataset, tokenizer, compute budget, and training setup as the 1.1B DLM.

Finetuning (8B model alignment). Since LLaDA-8B's pretraining data are unavailable, we compare memorization after a shared Enron adaptation protocol, including LLaMA-3.1-8B as a scale-matched ARM baseline. All DLMs and ARMs are finetuned for one epoch on Enron using each model's end-of-pretraining learning rate, ensuring that PII leakage comparisons are not driven by mismatched finetuning procedures.

Data and evaluation. For validation experiments, we use 100-token SlimPajama windows with random masks. For PII extraction, we use Enron emails and phone numbers extracted by regular expressions, using the preceding 100 tokens as the prefix. We filter email prompts that contain additional email addresses to reduce cue leakage. Full dataset, masking, and decoding details are consolidated in Appendix N, with extraction patterns in Appendix G.

6 Analysis and Results

This section empirically evaluates our memorization framework. We first validate that generalized discoverable extraction is measurable in DLMs despite stochastic masking and iterative denoising, and then confirm that finer sampling resolution increases verbatim memorization. We further align DLMs and ARMs in a PII completion setting to compare memorization-based leakage under identical prefix-conditioned reconstruction, and finally show the metric reflects memorization rather than generalization via train vs. disjoint same-domain test comparisons.

6.1 Rationality of generalized discoverable extraction in DLMs

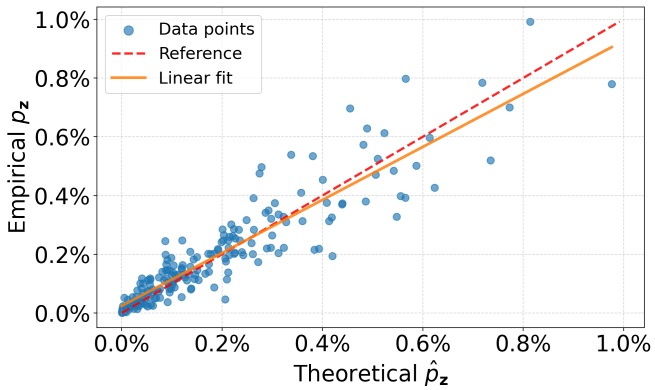


Figure 2: Empirical vs. theoretical memorization probability measured by DLM-1.1B on 200 SlimPajama examples. Empirical $p_{\boldsymbol{z}}$ is estimated from 100,000 random-decoding generations; theoretical $\hat{p}_{\boldsymbol{z}}$ is estimated from 1,024 mask-patterns at a fixed mask ratio ($[0.2, 0.3]$).

To validate Eq. 7, we compare the theoretical estimate $\hat{p}_z$ with a brute-force empirical estimate p_z on 200 SlimPajama examples using DLM-1.1B. Empirical probabilities are estimated from 100,000

stochastic decoding trials, while $\hat{p}_z$ is computed using 1,024 random masks. Figure 2 shows that the two estimates closely agree across the tested range, indicating that Eq. 7 provides a query-efficient approximation to brute-force decoding. Additional results under different numbers of mask-patterns are provided in Appendix D.

6.2 Impact of Sampling Resolution on Verbatim Memorization

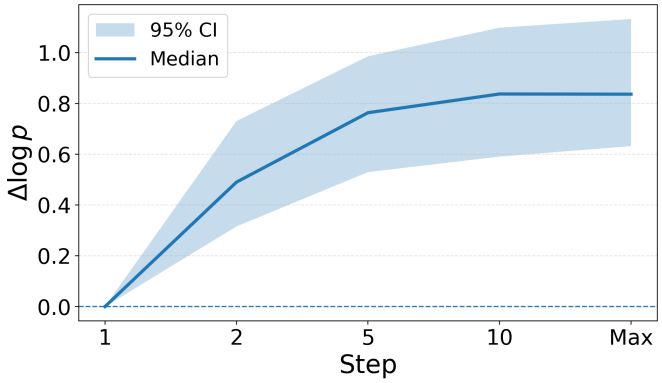


Figure 3: Empirical log-probability ratio across decoding steps (DLM-1.1B). We sample 200 mask-patterns and, for each, run 10,000 random decoding generations to estimate success probabilities at steps $\{1, 2, 5, 10, \text{per-token}\}$. We plot the median log-probability difference to one-step (baseline), with 95% bootstrap Confidence Intervals (CIs) across generation-trajectories.

In this section, we provide empirical evidence supporting Remark 4.2. Building on Sec 6.1, we select 200 mask-patterns from the previous section whose theoretical one-step recovery rate exceeds 0.01, a subset chosen for computational feasibility. For each selected trajectory, we run 10,000 unconstrained generation trials under five generation resolutions: 1-step ($N = 1$), 2-step ($N = 2$), 5-step ($N = 5$), 10-step ($N = 10$), and Max ($N = |\mathcal{M}|$, where the number of masked tokens is used as the generation step.)

Figure 3 summarizes these *empirical* exact-recovery success rates by plotting, for each resolution, the cross-trajectory median of the log success-rate ratio relative to the one-step baseline. We report $\Delta \log p := \text{median}\big(\log(\hat{p}_N/\hat{p}_1)\big)$, where $\hat{p}_N$ is the empirical exact-recovery hit rate at resolution N and the median is taken over the mask-patterns samples. The resulting curve exhibits a clear upward trend as the number of generation steps increases, suggesting an approximately monotonic relationship between generation resolution and exact-recovery success. This implies that allocating more denoising steps consistently increases the likelihood of reproducing the target content exactly, providing empirical support for Remark 4.2. Additional memorization examples are provided in Appendix F.

6.3 Empirical Analysis of PII Leakage via Aligned ARM–LDM Measurement

Under the aligned PII prefix-suffix verbatim completion task, DLMs exhibit substantially lower (n, p)-discoverable PII memorization than scale-matched ARMs. Table 1 reports the number of memorized email and phone samples at target probabilities $p \in \{0.5, 0.99\}$ under a query budget of $n = 10{,}000$ over 3,000 prompts. This gap may partly reflect differences between training objectives: random masking in DLMs disrupts learning to preserve long contiguous context, while the ARMs training paradigm naturally matches the sequential prefix-suffix decoding.

At a larger scale, the LLaDA-8B model exhibits a clear increase in memorization when moving from single-step reconstruction to per-token reconstruction, providing practical task evidence for Remark 4.2 that finer-grained denoising increases exact recovery probability. However, even under per-token reconstruction, the 8B diffusion model is only comparable to the 1.1B ARM. Importantly, this comparison should be interpreted in light of training cost: LLaDA-8B is trained with roughly two orders of magnitude more FLOPs and about $150\times$ more training token exposure than the 1.1B ARM baseline (15.6B vs. 2300B) Nie et al. [2025b]. Overall, in this aligned prefix-conditioned PII

Table 1: Generalized (n, p)-discoverable Extraction. Number of memorized samples at target probabilities p under a query budget of $n = 10{,}000$ on 3,000 completion prompts per PII type. All results report mean $\pm$ standard deviation across four fine-tuning seeds.

Model	Step	Email		Phone	
		$p = 50\%$	$p = 99\%$	$p = 50\%$	$p = 99\%$
DLM-1.1B	One	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
	Max	15.75 ± 0.96	6.50 ± 1.29	0.00 ± 0.00	0.00 ± 0.00
ARM-1.1B	N/A	212.50 ± 8.58	136.50 ± 4.80	4.00 ± 0.82	3.00 ± 0.00
LLaDA-8B	One	9.25 ± 0.50	8.00 ± 0.00	7.00 ± 0.00	3.25 ± 0.50
	Max	181.25 ± 1.71	121.75 ± 1.26	24.00 ± 1.15	17.00 ± 0.00
LLaMA-3.1-8B	N/A	366.50 ± 2.38	306.75 ± 2.99	650.75 ± 7.72	518.25 ± 9.18

setting, DLMs exhibit lower memorization behavior. Additional scale-wise PII results are reported in Appendix E.

The scale-matched LLaMA-3.1-8B baseline also exhibits higher PII extractability than LLaDA-8B under the same Enron adaptation and prefix-conditioned evaluation protocol. Tokenization is a potential contributor to the particularly large phone-number gap: phone targets average approximately 6 tokens under the LLaMA tokenizer, compared with 13 for the other models.

6.4 Validating Memorization Beyond Generalization

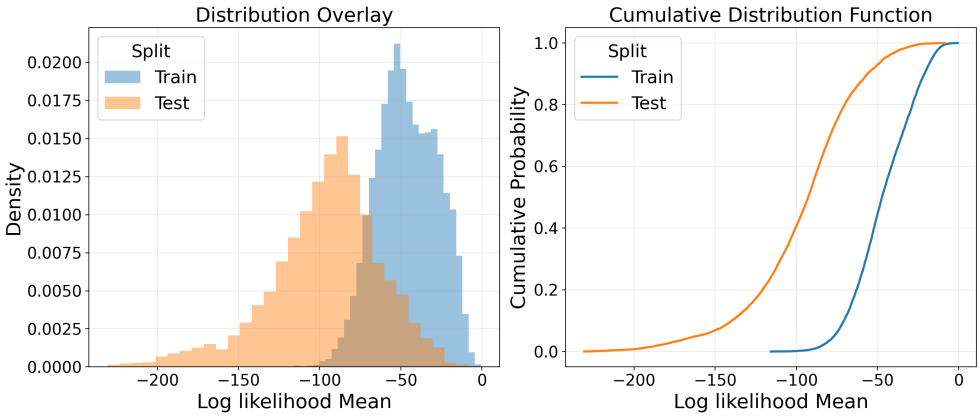


Figure 4: Validating memorization of training data. We compare the reconstruction likelihood from the Enron (train) against the unseen from the TREC 2007 Spam (test), which is from the same domain.

To distinguish memorization from generalization, we follow prior work Hayes et al. [2025] and compare Enron training data with the TREC 2007 Spam email dataset Bratko et al. [2006], a same-domain corpus disjoint from training. From each corpus, we sample 10,000 100-token windows and evaluate extraction at mask ratio $r = 0.25$ with 512 sampling queries per example.

Figure 4 shows the resulting log-likelihood distributions and CDFs. The Enron training distribution is shifted toward higher reconstruction likelihood than the unseen TREC distribution, indicating that the metric is sensitive to training exposure rather than merely reconstructing email-like text. Thus, although DLM-1.1B exhibits low overall verbatim extraction, its systematic train–test gap supports interpreting recovered Enron content as memorized training data rather than coincidental matches. Additional results for other model scales are provided in Appendix E, Figure 9.

6.5 Downstream Capability Comparison

To assess whether the lower PII extraction observed for DLM-1.1B simply reflects uniformly weaker model capability, we evaluate the 1.1B DLM and ARM checkpoints on five downstream benchmarks. These models share the parameter scale, pretraining dataset, tokenizer, and compute budget described in Section 5.

Table 2: Downstream benchmark scores (%) for the 1.1B ARM and DLM checkpoints. Higher is better.

Model	OpenBookQA	BoolQ	SocialIQA	RACE-high	LAMBADA
ARM-1.1B	30.10	55.75	37.46	28.71	30.57
DLM-1.1B	25.80	62.17	34.03	28.52	32.87

Table 2 shows mixed relative performance: the DLM scores higher on BoolQ and LAMBADA, whereas the ARM scores higher on OpenBookQA, SocialIQA, and marginally on RACE-high. These results do not indicate a uniformly weaker DLM and support using these controlled checkpoints for the memorization comparison. Complementary membership-inference results are reported in Appendix L; Appendix M examines how sampling resolution affects description-to-name task performance.

7 Conclusion and Future Work

This paper introduces a generalized probabilistic extraction framework for measuring memorization in diffusion language models, allowing discoverable extraction beyond prefix-based decoding to arbitrary masking patterns and stochastic diffusion sampling trajectories. It further establishes a principled link between sampling design and leakage risk by proving that finer-grained sampling resolution (more denoising steps) increases the probability of memorization, with autoregressive decoding recoverable as a special case of maximal resolution. Empirically, across model scales and sampling strategies, and under an aligned prefix-conditioned PII completion task, DLMs exhibit lower memorization than ARM counterparts. Building on these results, future work will measure how post-training (SFT and preference optimization) changes memorization in diffusion language models using evaluation protocols that preserve aligned behavior and match real-deployment prompting. In particular, it is important to test whether post-training drives DLMs toward more prefix–suffix-style generation, and whether this shift increases verbatim memorization and extractability.

Acknowledgements

We thank the Aalborg University AI:X initiative for enabling this work via the AI:SECURITY lab. JB and was further supported by the Novo Nordisk Foundation under the Ascending Data Investigator programme (NNF24OC0092972), the Independent Research Fund Denmark under the Sapere Aude programme (5254-00035B), and Coefficient Giving under the Technical AI Safety Research programme. We further acknowledge the support of the AAU AI Cloud and express our gratitude to DeiC for providing computing resources on the LUMI cluster (project nr. 465002249).

References

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

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

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion language modeling by estimating the ratios of the data distribution. 2023.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37: 103131–103167, 2024.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *arXiv preprint arXiv:2406.03736*, 2024.

Shen Nie, Fengqi Zhu, Chao Du, Tianyu Pang, Qian Liu, Guangtao Zeng, Min Lin, and Chongxuan Li. Scaling up masked diffusion models on text. In *ICLR*, 2025a.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, JUN ZHOU, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, pages 2633–2650, 2021.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*, 2022.

Hirokazu Kiyomaru, Issa Sugiura, Daisuke Kawahara, and Sadao Kurohashi. A comprehensive analysis of memorization in large language models. In Saad Mahamood, Nguyen Le Minh, and Daphne Ippolito, editors, *Proceedings of the 17th International Natural Language Generation Conference*, pages 584–596, Tokyo, Japan, September 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.inlg-main.45.

Jing Huang, Diyi Yang, and Christopher Potts. Demystifying verbatim memorization in large language models. *arXiv preprint arXiv:2407.17817*, 2024.

Bo Li, Qinghua Zhao, and Lijie Wen. Rome: Memorization insights from text, logits and representation. *arXiv preprint arXiv:2403.00510*, 2024.

Jie Zhang, Qinghua Zhao, Lei Li, and Chi-ho Lin. Extending memorization dynamics in pythia models from instance-level insights. *arXiv preprint arXiv:2506.12321*, 2025a.

Xiaoyu Luo, Yiyi Chen, Johannes Bjerva, and Qiongxiu Li. Shared path: Unraveling memorization in multilingual llms through language similarities. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Suzhou, China, 2025.

Jie Huang, Hanyin Shao, and Kevin Chen-Chuan Chang. Are large pre-trained language models leaking your personal information? *arXiv preprint arXiv:2205.12628*, 2022.

Siwon Kim, Sangdoo Yun, Hwaran Lee, Martin Gubri, Sungroh Yoon, and Seong Joon Oh. Propile: Probing privacy leakage in large language models. *Advances in Neural Information Processing Systems*, 36:20750–20762, 2023.

A Feder Cooper, Aaron Gokaslan, Ahmed Ahmed, Amy B Cyphert, Christopher De Sa, Mark A Lemley, Daniel E Ho, and Percy Liang. Extracting memorized pieces of (copyrighted) books from open-weight language models. *arXiv preprint arXiv:2505.12546*, 2025.

Ahmed Ahmed, A Feder Cooper, Sanmi Koyejo, and Percy Liang. Extracting books from production language models. *arXiv preprint arXiv:2601.02671*, 2026.

Jamie Hayes, Marika Swanberg, Harsh Chaudhari, Itay Yona, Ilia Shumailov, Milad Nasr, Christopher A Choquette-Choo, Katherine Lee, and A Feder Cooper. Measuring memorization in language models via probabilistic extraction. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 9266–9291, 2025.

AI@Meta. Llama 3 model card. 2024. URL https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

OpenAI. Hello GPT-4o. https://openai.com/index/hello-gpt-4o/, 2024.

Tiwei Bie, Maosong Cao, Kun Chen, Lun Du, Mingliang Gong, Zhuochen Gong, Yanmei Gu, Jiaqi Hu, Zenan Huang, Zhenzhong Lan, et al. Llada2.0: Scaling up diffusion language models to 100b. *arXiv preprint arXiv:2512.15745*, 2025.

Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. Analyzing leakage of personally identifiable information in language models. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 346–363. IEEE, 2023.

Xiaoyu Luo, Yiyi Chen, Qiongxiu Li, and Johannes Bjerva. Do llms really memorize personally identifiable information? revisiting pii leakage with a cue-controlled memorization framework. *arXiv preprint arXiv:2601.03791*, 2026.

Milad Nasr, Javier Rando, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from aligned, production language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

Andrej Bratko, Bogdan Filipi$c$, Gordon V Cormack, Thomas R Lynam, and Bla$z$ Zupan. Spam filtering using statistical data compression models. *The Journal of Machine Learning Research*, 7: 2673–2698, 2006.

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*, 2018.

Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Frank Yang, and Hai Li. Min-k%++: Improved baseline for pre-training data detection from large language models. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=ZGkfoufDaU.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Please see the Abstract, Sec. 1, Sec. 4, and Sec. 6.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please see Appendix H

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Please see Sec. 4.1.2 and Appendix A.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Please see Sec. 5, Secs. 6.1–6.4, Appendix G

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Please see the supplementary code and Appendix G

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Please see Sec. 5, Secs. 6.1–6.4, Appendix B, Appendix G

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Please see Fig. 3 and Sec. 6.2; full count/distribution results are reported in Table 1 and Fig. 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please see Appendix K

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Please see Sec. 6.3 and Appendix **??**.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Please see Appendix I

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Please see Appendix J.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Please see the supplementary code.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs were used only for writing assistance, such as grammar correction and language polishing

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

A Proof of Theorem 4.3

Proof. Consider two sampling step counts $N_1 > N_2$, where the partition $\mathcal{P}_{N_1}$ is a refinement of $\mathcal{P}_{N_2}$. For any token $\pi \in \mathcal{M}$, let $\mathcal{U}^{(prev)}$ and $\mathcal{W}^{(prev)}$ be the observed index sets at the step immediately preceding the recovery of π in the respective N_1 and N_2 steps sampling processes.

Since $\mathcal{P}_{N_1}$ is a refinement of $\mathcal{P}_{N_2}$ and the recovery order σ is fixed, the N_1 process updates the observed set more frequently. Specifically, for any token π, the set of tokens already recovered in the N_1 process is always a superset of those recovered in the N_2 process at the time π is sampled:

$$\mathcal{W}^{(prev)} \subseteq \mathcal{U}^{(prev)}. \tag{12}$$

Since the compared contexts consist of correct tokens along the exact-recovery event, the correct-context monotonicity property in Remark 4.2 applies. Therefore, for every $\pi \in \mathcal{M}$:

$$\Pr(\hat{z}_\pi = z_\pi \mid \boldsymbol{z}_{\mathcal{U}^{(prev)}}) \geq \Pr(\hat{z}_\pi = z_\pi \mid \boldsymbol{z}_{\mathcal{W}^{(prev)}}). \tag{13}$$

Since we assume the same recovery order of tokens in two cases, taking the product over all $\pi \in \mathcal{M}$, we can conclude that:

$$\Pr(\hat{\boldsymbol{z}}_{\mathcal{M}} = \boldsymbol{z}_{\mathcal{M}} \mid \boldsymbol{z}_{\bar{\mathcal{M}}}, N_1) \geq \Pr(\hat{\boldsymbol{z}}_{\mathcal{M}} = \boldsymbol{z}_{\mathcal{M}} \mid \boldsymbol{z}_{\bar{\mathcal{M}}}, N_2). \tag{14}$$

This completes the proof. □

B ARMs and DLMs Sampling Paradigm

We then briefly introduce the sampling paradigms that are commonly used in ARMs and DLMs. Among these paradigms, temperature and top-k sampling are widely adopted in ARMs Fan et al. [2018], while DLMs use Gumbel noise perturbation Nie et al. [2025a].

Temperature Sampling. Given a prefix $\boldsymbol{z}^{1:l-1}$, let ℓ_v denote the logit for token $v \in \mathcal{V}$. Temperature sampling parameterizes the predictive categorical distribution as

$$\Pr(z^l = v \mid \boldsymbol{z}^{1:l-1}) = \frac{\exp(\ell_v/T)}{\sum_{u\in\mathcal{V}} \exp(\ell_u/T)}, \quad T > 0.$$

Here, T serves as a scaling factor that controls the sharpness of the distribution: a lower T concentrates the probability mass on high-logit tokens, thereby reducing stochasticity, while a higher T flattens the distribution, promoting diversity in generation. The standard softmax distribution is recovered as a special case when $T = 1$.

Top-k sampling. Let $p(v) = \mathrm{softmax}(\ell_v)$ and let $\mathcal{K}$ be the set of k tokens with largest $p(v)$. Top-k sampling restricts the distribution to $\mathcal{K}$ and renormalizes it as

$$Pr(z^l = v \mid \boldsymbol{z}^{1:l-1}) \;=\; \begin{cases} \frac{p(v)}{\sum_{u\in\mathcal{K}} p(u)}, & v \in \mathcal{K}, \\ 0, & \text{otherwise.} \end{cases}$$

This procedure collapses to deterministic greedy decoding when $k = 1$, where only the most probable token is selected.

Gumbel noise perturbation. In diffusion models, a commonly used sampling paradigm is Gumbel noise perturbation. To perform stochastic token sampling, we draw i.i.d. random variables $U_v \sim \mathrm{Uniform}(0, 1)$ for each vocabulary item $v \in \mathcal{V}$, and construct the perturbed scores as

$$S_v \;=\; \frac{\exp(\ell_v)}{\big(-\log U_v\big)^{T_{\mathrm{Gumbel}}}}, \tag{15}$$

where T_{Gumbel} is a temperature parameter that controls the level of stochasticity. The sampled token is then selected as the most confident one, i.e.,

$$\arg\max_{v\in\mathcal{V}} S_v.$$

When $T_{\mathrm{Gumbel}} = 0$, $S_v = \exp(\ell_v)$. In this case, the procedure reduces to greedy decoding, which can be naturally paired with deterministic sampling in diffusion models.

C Validation of generalized (ϵ, n, p)-discoverable extraction

We initiate our empirical analysis by performing $R = 10,000$ stochastic generation trials on a series of infilling tasks, where 20 tokens are randomly masked within a 100-token window, randomly selecting from Sec 6.2 masked-pattern. For each trial, we record the exact number of wrongly reconstructed tokens. Based on the empirical observations of the resulting distribution, we postulate the following assumptions.

Assumption C.1. For a diffusion-based LLM, let $d(\hat{\boldsymbol{z}}^{(i)}_{\mathcal{M}}, \boldsymbol{z}_{\mathcal{M}})$ denote the Hamming distance between the i-th independently sampled reconstruction and the ground-truth suffix. We observe that for sufficiently large $|\mathcal{M}|$, the distribution of d over multiple trials exhibits a strong central tendency. Specifically, within its central regime (excluding the tails), the distribution can be approximated by a normal distribution

$$d(\hat{\boldsymbol{z}}^{(i)}_{\mathcal{M}}, \boldsymbol{z}_{\mathcal{M}}) \approx \mathcal{N}(\mu, \sigma^2), \tag{16}$$

where the mean μ and variance σ^2 are parameters governed by the model's intrinsic extraction capability.

This Gaussian behavior is motivated by the fact that the Hamming distance can be decomposed into a sum of M indicator random variables

$$d = \sum_{j=1}^{|\mathcal{M}|} \mathbb{I}(\hat{z}_{\pi_j} \neq z_{\pi_j}), \tag{17}$$

where each indicator represents a token-level recovery failure. Owing to the inherent stochasticity of the sampling process in large DLMs, the total error count can be modeled as a sum of weakly dependent Bernoulli random variables. In accordance with the Central Limit Theorem (CLT) for weakly dependent sequences, the aggregate sum converges toward a normal distribution as the number of masked tokens $|\mathcal{M}|$ increases. The Quantile–Quantile (Q-Q) plots in Figure 5 (left) provide strong empirical evidence for the normality of the recovered token counts. Specifically, the strict adherence of the sample quantiles to the diagonal reference line suggests that the error distribution across $10,000$ trials is well-characterized by a normal distribution. While minor deviations appear in the tail portions, the empirical distribution aligns well with the theoretical normal curve in the central regime.

In contrast, Figure 6 reveals a markedly different trend for ARMs. Due to the sequential nature of the ARM sampling process, there exist significantly stronger inter-token dependencies, which prevent the distribution from converging to a normal form, leading to a distinct departure from the normality assumption.

The right column of Figure 5 compares the empirical CDF obtained via (11) with the Gaussian CDFs fitted using $10,000$ and 128 trials, respectively. As illustrated, the Gaussian approximation aligns well with the empirical CDF, effectively capturing the behavior of $\hat{p}_{\mathbf{z},\epsilon}$. While minor discrepancies still appear in the extreme regions, where the Gaussian model tends to underestimate the probability of tail events far from the mean, the overall fit remains robust. Notably, the curves fitted with 128 and $10,000$ trials are remarkably similar. This suggests that a significantly smaller number of trials can be employed to obtain a rough estimation of $\hat{p}_{\mathbf{z},\epsilon}$ within the central regime, offering a more query-efficient alternative to the empirical counting method

$$\hat{p}_{\mathbf{z},\epsilon} = \Pr(d(\hat{\mathbf{z}}^{(i)}_{\mathcal{M}}, \mathbf{z}_{\mathcal{M}}) \leq \epsilon) \approx \Phi\left(\frac{\epsilon - \mu}{\sigma}\right), \tag{18}$$

where $\Phi(\cdot)$ denotes the CDF of the standard normal distribution $\mathcal{N}(0,1)$, while μ and σ represent the mean and standard deviation estimated from a few trials.

D Effect of Masked-patterns on Memorization Estimation

In this section, we provide additional empirical evidence regarding the impact of the number of sampled masked-patterns R on the estimation of memorization probabilities. Figure 7 illustrates the correlation between the empirical memorization probability (p_z) and the theoretical prediction ($\hat{p}_z$) across different masked-patterns.

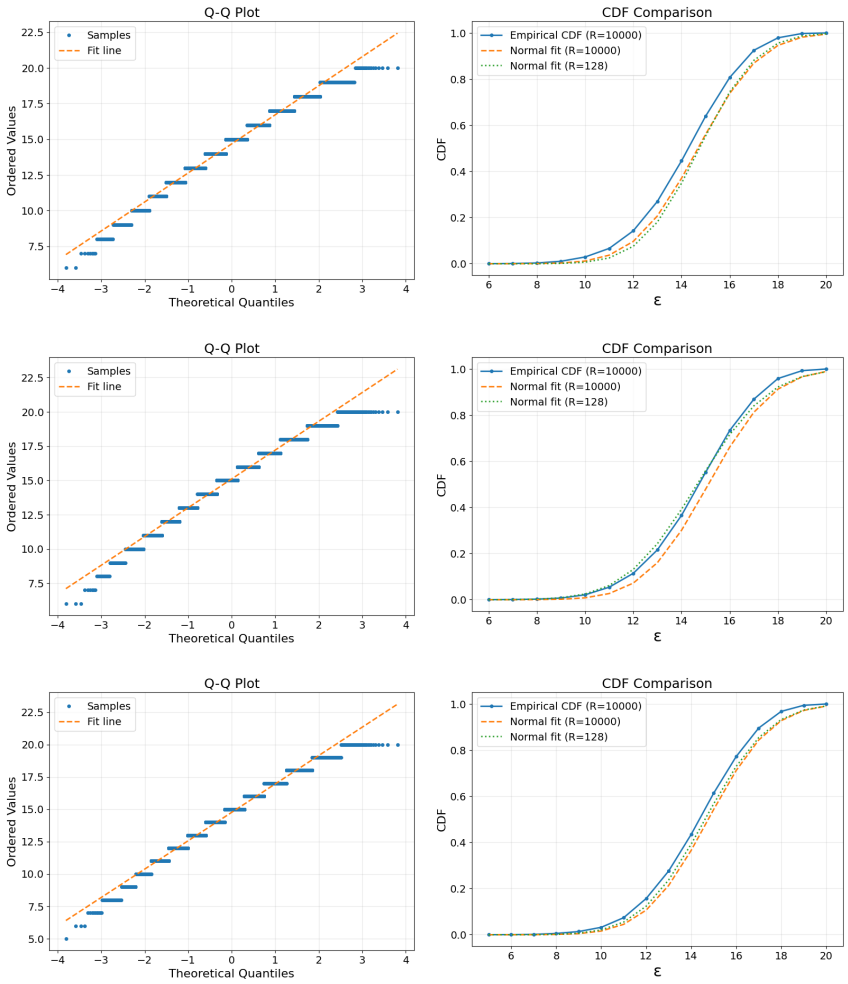


Figure 5: Paired Q-Q plots (left) and CDF comparisons (right) for three cases for DLM.

Convergence and Accuracy. Our results demonstrate that the estimation accuracy significantly improves as the number of sampled masked-patterns R increases from 128 to 512. We observe that sampling $R = 512$ times provides a high-fidelity fit to the theoretical line, suggesting that this amount of masked-patterns is sufficient to capture the underlying memorization behavior of DLMs without prohibitive computational overhead.

Analysis of Empirical Variance. We observe that the empirical points exhibit a degree of dispersion around the identity line across the entire probability spectrum . This variance is a natural consequence of the stochastic nature of the denoising process in DLMs:

- **Stochastic Sampling Fluctuations:** Since each empirical data point is calculated as the mean of R random masked, it is essentially a realization of a binomial-like estimator. Therefore, some level of statistical fluctuation is inevitable. Even for samples with high theoretical memorization probability, the specific set of R trajectories sampled may include more or fewer "successful" extraction paths than the expected average.
- **Natural Statistical Dispersion:** This variance does not represent a flaw in the theoretical derivation, but rather reflects the inherent randomness of the sampling-based estimation. Crucially, as R increases, the standard deviation of these fluctuations decreases, and the empirical values remain centered around the theoretical identity line within a consistent order of magnitude.

In summary, the empirical results in Figure 7 confirm that our theoretical framework provides an accurate "mean" behavior for data extraction. The observed variance is consistent with standard

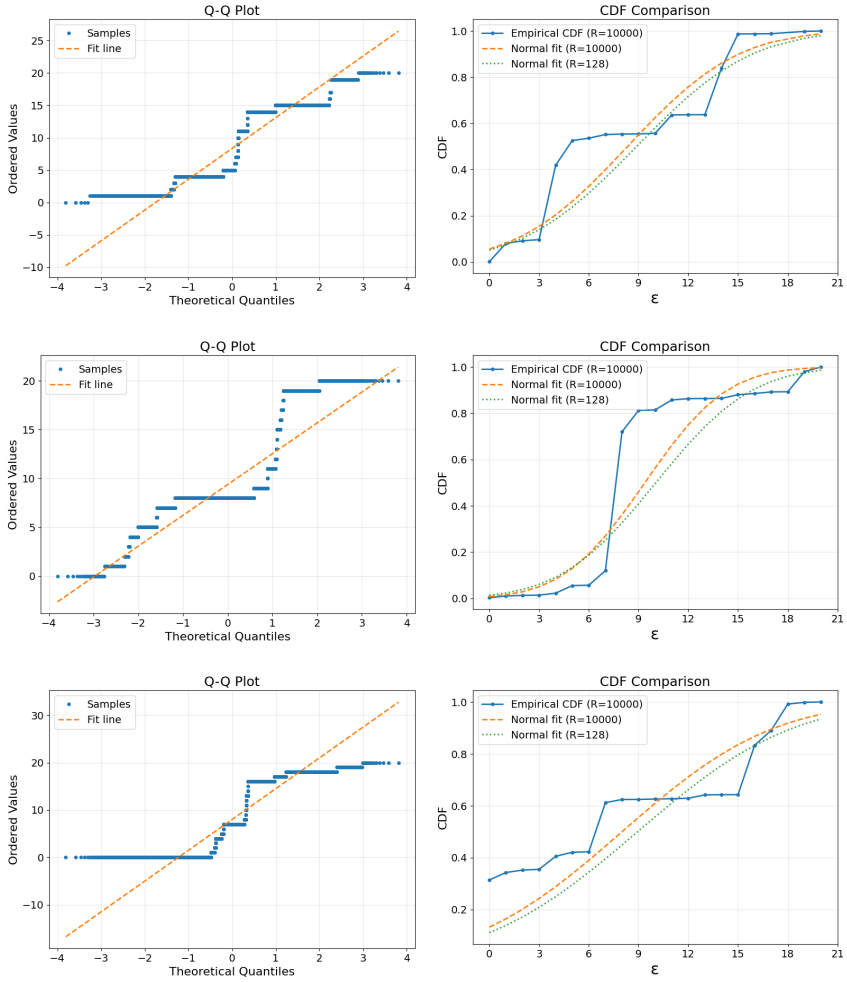


Figure 6: Paired Q-Q plots (left) and CDF comparisons (right) for three cases for ARM.

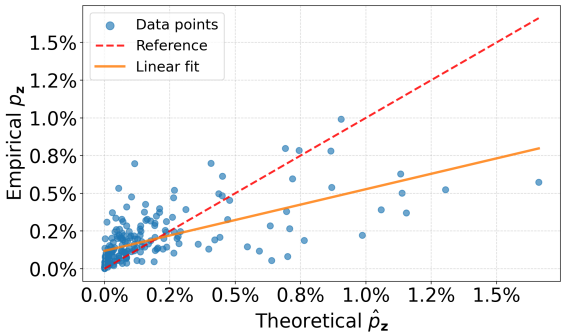
Monte Carlo estimation errors and does not impede the practical utility of our metric in identifying memorized content.

E PII Memorization

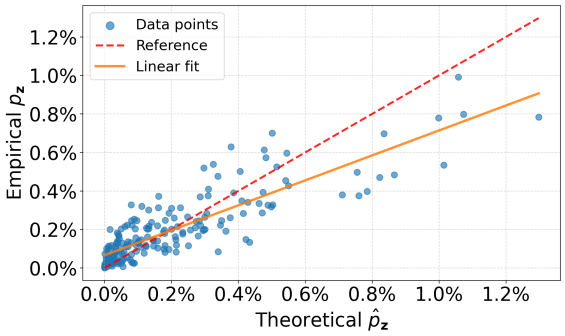
In this section, we extend our evaluation of (n, p)-discoverable memorization to smaller model variants (DLM-170M and DLM-629M) across different types of Personally Identifiable Information (PII). Table 3 summarizes the number of memorized samples for email addresses and phone numbers under varying target probability thresholds p.

Scale-wise DLM PII leakage. Figure 8 reports (n, p)-discoverable email memorization across DLM scales. The extractable rate increases with model size, but the gains from larger query budgets are sublinear: most practically discoverable email memorization is recovered with modest n, while further increasing n yields limited additional leakage. Table 3 reports the corresponding results for additional PII types and model scales.

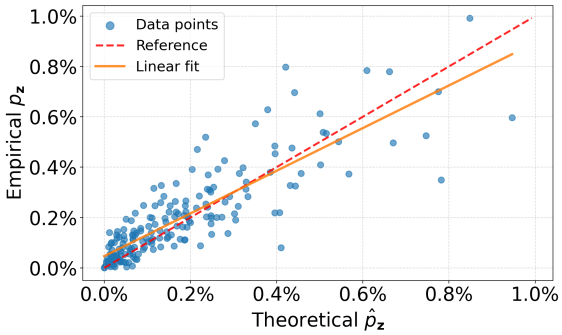
Low Leakage in Small-Scale DLMs. Our empirical results indicate that small-scale DLMs exhibit extremely low leakage probabilities for PII. This is particularly evident for *phone numbers*, where the number of memorized samples remains zero across all tested models and confidence levels $p \in \{10\%, 50\%, 90\%, 99\%\}$. Even for email addresses, which appear slightly more susceptible to extraction, the risk remains significantly lower compared to larger model scales.



(a) masked-patterns = 128



(b) masked-patterns = 256



(c) masked-patterns = 512

Figure 7: Comparison of empirical and theoretical memorization probabilities under different sampling masked-patterns quantities. As the number of samples increases, the empirical values converge towards the theoretical identity line.

Validation of Model Memorization. To ensure that these low extraction rates are a meaningful reflection of the model's privacy-preserving properties—rather than a byproduct of underfitting or a general failure to memorize the training set—we conducted auxiliary measurements. Specifically, we evaluated the DLM-170M and DLM-629M models using the same training data reconstruction methodology described in Sec 6.4. Figure 9 confirmed that both models show higher confidence in the Enron training corpus. Therefore, the low extraction rates reported in Table 3 serve as a valuable and reliable reference for PII risks in smaller diffusion language models.

F Memorization examples

We include the following memorized examples from Section 6.2 in this appendix.

Model	Step	Email				Phone			
		$p = 10\%$	$p = 50\%$	$p = 90\%$	$p = 99\%$	$p = 10\%$	$p = 50\%$	$p = 90\%$	$p = 99\%$
DLM-170M	One	0	0	0	0	0	0	0	0
	Max	7	2	0	0	0	0	0	0
DLM-629M	One	0	0	0	0	0	0	0	0
	Max	17	10	7	4	0	0	0	0
DLM-1.1B	One	0	0	0	0	0	0	0	0
	Max	26	16	10	7	0	0	0	0

Table 3: Generalized (n, p)-discoverable Extraction. Number of memorized samples at target probabilities p under a query budget of $n = 10{,}000$ on $3{,}000$ completion prompts, for different PII types.

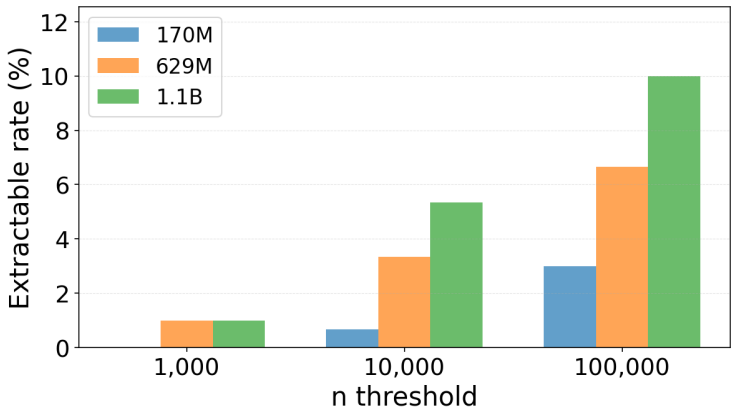


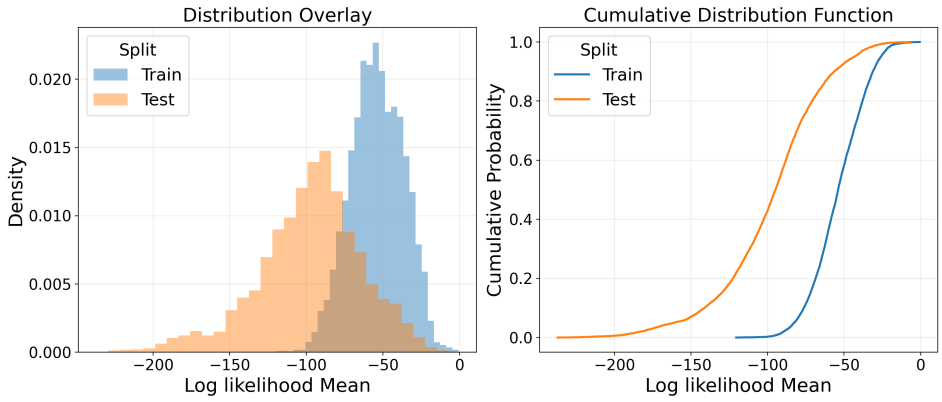
Figure 8: Per-token-step email hit rates under different model scales (170M/629M/1.1B). Each bar reports the Extractable rate over $3{,}000$ samples, measured as whether an email can be recovered within a querying budget n, at fixed $p = 0.5$.

Original

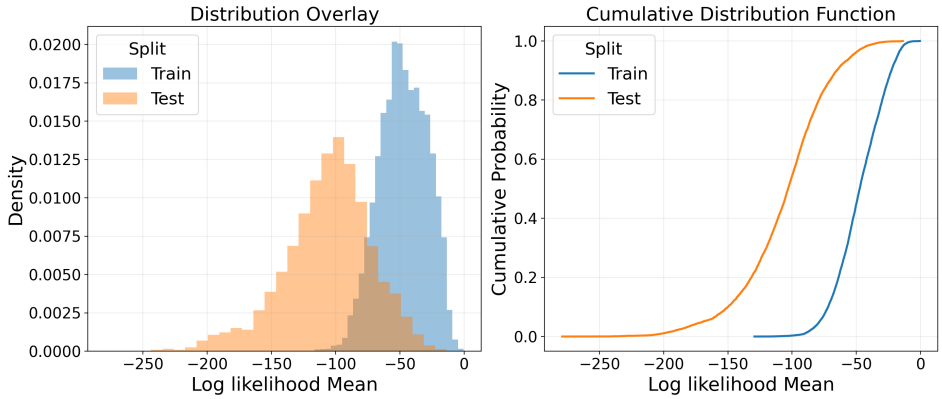
```
1948 Summer Olympics.

References

External links


1927 births
2020 deaths
Athletes (track and field) at the 1948 Summer Olympics
Brazilian female sprinters
Olympic athletes of Brazil
Olympic female sprinters</s>  Why is Tommy Hunter making us all nervous?
One of the Phillies' offseason acquisitions
```

Recoverable Masked-Pattern

```
<MASK>  <MASK> 948 <MASK>  <MASK> . <MASK>
<MASK> ferences

External <MASK>


<MASK> 92 <MASK>  <MASK> s <MASK>  <MASK> 020 deaths <MASK> Athlet <MASK>  (track and <MASK> ) at the
19 <MASK>  <MASK>  Summer <MASK>  <MASK> Brazilian female sprinter <MASK>
Olympic <MASK> es <MASK>  Brazil <MASK> O <MASK> ic <MASK>  spr <MASK> s</s>  Why <MASK>  Tommy Hunter
making us all nervous?
One <MASK>  <MASK>  Phillies' offseason <MASK> quisitions
```



(a) DLM-170M



(b) DLM-629M

Figure 9: Validating memorization of training data under different DLM scales. We compare the reconstruction likelihood from the Enron (train) against the unseen from the TREC 2007 Spam (test), which is from the same domain.

Original

```
te change is a natural phenomenon caused by several factors such as solar output, the planet's orbit,
and ocean circulation. However, human activity that releases greenhouse gases into the atmosphere has
made the effects of climate change more severe.
According to scientists, even if the world stopped emitting carbon dioxide today, global warming would
persist for decades to come because of the amount of carbon dioxide we've already pumped into the
atmosphere.
```

Recoverable Masked-Pattern

```
te change is a natural <MASK>  <MASK>  caused <MASK>  several factors such <MASK>  solar output <MASK>
<MASK>  planet <MASK> s orbit, <MASK>  ocean circulation. However, human activity that releases green
<MASK>  gases <MASK>  the atmosphere <MASK>  made <MASK>  effects of climate change more severe.
According <MASK>  scientists, even <MASK>  the <MASK>  <MASK>  emitting carbon dioxide today, global
war <MASK>  would persist for decades to come because of the amount of carbon diox <MASK>  we've
already <MASK> ump <MASK>  into the atmosphere.
```

Original

```
control. Researchers believe that hereditary ataxia may supply clues to the genetic causes of RLS.
Attention-deficit hyperactivity disorder (ADHD).
Psychiatric disorders, such as depression.
Environmental and Dietary Factors
The following environmental and dietary factors can trigger or worsen RLS:
Iron deficiency. People who are deficient in iron are at risk for R
```

Recoverable Masked-Pattern

```
control. Researchers believe that her <MASK> ary ataxia may supply clues to the genetic causes of RLS.
Att <MASK>  <MASK>  <MASK> ic <MASK>  hyperactivity disorder <MASK> AD <MASK> ).
<MASK> sychiatric disorders, such as de <MASK>  <MASK>
<MASK> al and <MASK> ary Factors
The following environmental and <MASK> etary <MASK>  can trigger or <MASK> orsen RLS <MASK>
I <MASK>  deficiency. People who <MASK>  defic <MASK>  <MASK>  iron are at risk for R
```

G Dataset Collection

Validating Data Collection. For the validation experiments (Sec 6.1, 6.2), we draw evaluation samples from the pretrain dataset by uniformly sampling **100-token windows** at random from the corpus. For each window, we apply random masking with mask ratio $r \in \{0.20, 0.25, 0.30\}$, and evaluate reconstruction/extraction under these different masking levels.

PII Collection. For the alignment experiment (Sec. 6.3), we construct a PII-focused evaluation set from the Enron email dataset, extracting emails and phone numbers via regular expressions, following Kim et al. [2023] procedure. We sample **3,000** email-containing texts and **3,000** phone-number-containing texts, and for each PII occurrence use the 100 preceding tokens as the prefix to test **verbatim memorization** under prefix-conditioned generation. To avoid spurious pattern-based cues, email prompts are further filtered to contain no additional email addresses.

We adopt the regular expressions in Kim et al. [2023]. to search for email addresses and U.S. phone numbers; the patterns are shown below.

```
EMAIL_RE = re.compile("^([a-zA-Z0-9_\-\.]+)@([a-zA-Z0-9_\-\.]+)\.([a-zA-Z]{2,5})$")
PHONE_RE = re.compile("[0-9][0-9][0-9][-.()][0-9][0-9][0-9][-.()
    ][0-9][0-9][0-9][0-9]")
```

H Limitations

Our DLM–ARM comparison is limited to an aligned prefix-conditioned PII completion setting. This setting enables controlled comparison with ARMs, but does not exhaust native DLM use cases such as bidirectional infilling. In addition, LLaDA-8B's original pretraining data are not publicly available, so we compare it after a shared Enron adaptation protocol.

Our experiments mainly study memorization in pretraining or continued-pretraining settings. We do not systematically evaluate how post-training procedures, such as instruction tuning, supervised fine-tuning, or preference optimization, affect DLM memorization. The Enron adaptation used in our PII experiments is intended to create a controlled exposure setting, rather than to characterize deployment style SFT behavior.

I Impact Statements

This paper presents a theoretical and empirical analysis of memorization in diffusion-based Large Language Models. Our findings contribute to the broader field of trustworthy AI by elucidating how sampling strategies influence the risk of training data leakage. While our analysis reveals potential vulnerabilities in how these models retain training data, this understanding is a prerequisite for building robust defenses and ensuring the responsible deployment of generative models in privacy-sensitive domains. There are many potential societal consequences of our work, and we believe our contribution to data transparency and model auditing is the most significant one.

J Assets and Licenses

We use several existing datasets, models, and training resources. SlimPajama-627B is released by Cerebras under Apache-2.0, with its underlying data subsets subject to their respective licenses and terms. The Enron email corpus was made public by the Federal Energy Regulatory Commission during the Enron investigation and is distributed through the CMU CALO release; the CMU release does not specify a standard open-source license, so we use it only for research evaluation and report aggregate or redacted extraction results. The TREC 2007 Spam corpus is used as a public research corpus for same-domain train–test comparison, subject to the original corpus agreement; we did not identify a separate OSI-style license for this corpus. LLaDA-8B is released under the MIT license. The ARM/DLM training recipe follows the SMDM codebase from prior work, which is released under Apache-2.0. We cite the original sources in Secs. 2 and 5 and use all assets in accordance with their stated licenses or terms of use.

K Compute Resources

All reported experiments were conducted on AMD Instinct MI250X GPU. The reported experiments used approximately 800 gpu-hours in total.

L Membership Inference Evaluation

We complement the extraction experiments with membership inference attacks (MIAs) on the 1.1B ARM and DLM. Each evaluation uses 1,000 member and 1,000 non-member Enron windows of 128 tokens. We evaluate a likelihood (loss-based) attack, Min-K% Prob, and Min-K%++ [Zhang et al., 2025b], using $k = 20\%$ for the latter two methods. Table 4 reports results averaged over four independently trained seeds.

Table 4: Membership inference on Enron. AUROC and TPR at 1% FPR are reported in percent, averaged over four seeds.

Model	Attack	AUROC	TPR@1% FPR
ARM-1.1B	Likelihood (loss)	51.58	1.52
	Min-K% Prob ($k = 20\%$)	51.91	1.47
	Min-K%++ ($k = 20\%$)	52.54	1.30
DLM-1.1B	Likelihood (loss)	49.93	0.90
	Min-K% Prob ($k = 20\%$)	50.04	1.15
	Min-K%++ ($k = 20\%$)	50.03	0.97

Across the tested attacks, AUROC remains close to the chance level of 50%, and TPR at 1% FPR is close to the random baseline of 1%. Under this evaluation protocol, these attacks provide little separation between members and non-members and do not yield a strong signal for comparing memorization across the two model families.

M Sampling Resolution and Task Performance

We examine the effect of sampling resolution on a description-to-name (D→N) task using fictitious name–description relations, following the reverse-curse setting of Nie et al. [2025a]. In D→N Direct, the model is fine-tuned on description-to-name relations and evaluated in the same direction with an unseen prompt template. In D→N Reverse, fine-tuning exposes only name-to-description relations, and evaluation asks the model to recover a name from its description. Accuracy measures whether the target name appears in the generated response.

Starting from the pretrained 1.1B masked diffusion model, we fine-tune on the original 3,600 prompt–completion pairs for 10 epochs. We concatenate each prompt and completion, append an EOS token, and apply the standard masked diffusion corruption objective to the entire sequence. We compare one-step generation with maximum-resolution, per-token generation.

Table 5: Description-to-name task accuracy (%) under two sampling resolutions for the 1.1B DLM.

Sampling resolution	D→N Direct	D→N Reverse
One step	74.00	22.67
Max step	99.33	84.00

Higher sampling resolution improves accuracy in both settings. Together with the extraction results in Section 6.2, this illustrates a potential trade-off: finer-grained generation can improve task performance while also making memorized content easier to recover.

N Consolidated Experimental Settings

This appendix collects the training, data, decoding, and evaluation settings used throughout the paper. Dataset extraction patterns are provided in Appendix G; the sampling mechanisms are described in Appendix B.

N.1 Pretraining and Enron Adaptation

The 170M, 629M, and 1.1B DLMs and the 1.1B ARM are pretrained on SlimPajama using the same tokenizer and training pipeline, following Nie et al. [2025a]. The pretraining budget is approximately 10^{21} FLOPs per model. Table 6 lists the shared training configuration.

Table 6: Pretraining configuration for the controlled ARM/DLM models.

Setting	**Value**
Sequence length	2,048
Optimizer	AdamW
Peak / minimum learning rate	2×10^{-4} / 2×10^{-5}
Weight decay	0.1
Adam betas / epsilon	$(0.9, 0.95)$ / 10^{-8}
Micro-batch size	8 per GPU
Global batch size	192
Number of GPUs	6
Gradient accumulation	4 steps
Gradient clipping	Global norm 1.0
Learning-rate schedule	1% warmup, followed by cosine decay
Precision	bf16-mixed
Pretraining seed	3407

For the PII comparison, models undergo one epoch of continued training on Enron. The 1.1B ARM and DLM use a fixed learning rate, and the 8B LLaDA and LLaMA models also use fixed learning rates during adaptation.

N.2 Data Processing and Conditioning

For Sections 6.1 and 6.2, we sample contiguous 100-token windows from SlimPajama and apply mask ratios $r \in \{0.20, 0.25, 0.30\}$. Unmasked tokens form the conditioning context, and masked positions form the reconstruction target. Section 6.1 uses 200 examples, 100,000 stochastic generations per empirical estimate, and 1,024 samples for the trajectory-likelihood estimate. Section 6.2 uses a fixed set of 200 mask patterns selected for a preliminary one-step recovery probability above 0.01. Each pattern is evaluated with 10,000 independent generations at each resolution in $\{1, 2, 5, 10, |\mathcal{M}|\}$. This selection targets patterns with measurable recovery probabilities; it is not a population-wide estimate of memorization prevalence.

For PII extraction, we sample 3,000 email-containing texts and 3,000 phone-number-containing texts from Enron. Emails and U.S. phone numbers are identified with the regular expressions in Appendix G. The raw 100-token context immediately preceding the PII forms the prompt, and the PII string forms the completion target. Email prompts containing additional email addresses are excluded. For DLM evaluation, all target suffix tokens are initially masked.

For the train–test comparison in Section 6.4, we sample 10,000 windows of 100 tokens from each of Enron and TREC 2007 Spam, using mask ratio $r = 0.25$ and 512 sampling queries per example. Membership inference uses the separate 128-token-window protocol in Appendix L.

N.3 Decoding and Probability Estimation

ARM suffix evaluation uses temperature $T = 1.0$, top-k truncation with $k = 40$, and evaluation batch size 32. The truncated token distribution is renormalized at each position. If a target token falls outside the top 40, its conditional probability, and hence the exact suffix probability, is zero. DLM evaluation uses temperature $T = 1.0$, no top-k truncation, and classifier-free guidance scale 0. In Section 6.1, generated tokens are sampled stochastically rather than selected greedily.

For an ARM, the single-query exact suffix probability is the product of the target-token conditional probabilities under the specified decoding distribution. For a DLM, the estimate is a probability-space average of conditional exact-recovery likelihoods over sampled recovery trajectories, as in Eq. 7.

Given the estimated single-query recovery probability $\hat{p}_{\boldsymbol{z}}$, independence of complete queries yields the estimated probability $1-(1-\hat{p}_{\boldsymbol{z}})^n$ of at least one successful recovery within n queries. Accordingly, an example is counted as (n,p)-discoverably extractable when

$$\hat{p}_{\boldsymbol{z}} \geq 1-(1-p)^{1/n}. \tag{19}$$

For $n = 10{,}000$, the thresholds are approximately 6.9312×10^{-5} at $p = 0.5$ and 4.6041×10^{-4} at $p = 0.99$. The PII tables count the examples meeting the corresponding threshold.

N.4 Additional Evaluations

Section 6.5 reports the five downstream benchmark scores for the controlled 1.1B checkpoints. Appendix L specifies the MIA sample sizes, attacks, and aggregation, while Appendix M describes the synthetic description-to-name data, ten-epoch adaptation, and name-recovery metric.