Extracting Training Data from Diffusion Language Models via Infilling

Yihan Wang *
University of Waterloo
yihan.wang@uwaterloo.ca

N. Asokan
University of Waterloo
KTH Royal Institute of Technology
asokan@acm.org

Abstract

Memorization in large language models has been studied almost exclusively through prefix-conditioned extraction, a natural choice for autoregressive models. However, diffusion language models (DLMs) can denoise masked tokens at arbitrary positions. Thus, prefix-only probing reveals only one facet of memorization in DLMs and significantly underestimates the risk of training-data extraction. In order to realistically model extractability of training data in DLMs, we introduce *infilling extraction*, a data-extraction protocol parameterized by an arbitrary binary mask that subsumes prefix-only probing and accounts for the bidirectional inductive bias of DLMs. Instantiating it on LLaDA-8B and Dream-7B across five extraction modes, three training pipelines, and three corpora covering verbatim and partial leakage, we find that mask geometry governs extractability: edge-conditioned masks *extract up to three times more* verbatim sequences than prefix-conditioned ones, and bidirectional access opens channels inaccessible in autoregressive models. In particular, an adversary who obtains a PII-redacted copy of the training data can use targeted infilling to raise PII recall on fine-tuned DLMs by 2.5–4.5× over prefix-conditioned extraction, substantially narrowing the gap to a scale-matched autoregressive model and, for LLaDA on email addresses, exceeding it under greedy prefix-conditioned extraction. Tunable parameters for decoding measurably affect extraction performance, while a follow-up supervised fine-tuning stage does not eliminate the prior memorization. Code is available at https://github.com/ssg-research/DLM-Infilling-Extraction.

1 Introduction

By learning statistical patterns over vast and diverse text, large language models (LLMs) exhibit surprising generalization ability, successfully adapting to a wide variety of downstream tasks. However, this reliance on training data comes at a cost: models do not merely generalize, but also memorize portions of their training data, raising serious concerns about the leakage of personally identifiable information (PII), the reproduction of copyrighted material, and the disclosure of confidential content embedded in training corpora [8, 26, 22, 32]. A prominent line of work demonstrating these risks is *data extraction attacks* [6, 35, 54, 18], in which an adversary crafts prompts that induce the model to regurgitate sequences from its training set.

While memorization and data extraction attacks on autoregressive language models (ARMs) have been studied extensively [51], a fundamentally different paradigm of text generation has recently emerged: diffusion language models (DLMs) [30, 17, 36, 53, 52]. Rather than predicting the next token strictly left-to-right, DLMs learn to iteratively denoise corrupted text and can generate multiple tokens at arbitrary positions per step [3, 31, 42, 19], offering a promising path toward faster and more

*Corresponding author.

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

flexible language models [41, 1, 24, 4, 5]. Despite their growing scale and deployment, memorization characteristics of DLMs and their vulnerability to data extraction remain largely unexplored.

In this work, we close this gap by developing a comprehensive framework for evaluating data extractability in large DLMs. Our central observation is that estimating the risk of training data extraction in DLMs using only prefixes from training data records as prompts [33] (also known as the *prefix-conditioned protocol*) can significantly underestimate the risk because this protocol, inherited from exploring data extraction in ARMs, captures only one facet of memorization: it asks what the model produces *next*, but not what it would fill in given *arbitrary surrounding context*. We therefore introduce *infilling extraction*, a strictly more general protocol parameterized by a binary mask $\mathbf{m}$. The unmasked positions of a training sequence constitute the context revealed to a DLM, which is asked to reconstruct the text corresponding to the masked positions. The training sequence is considered extracted if the reconstruction matches the masked text.

Within this protocol, we instantiate five canonical access modes which we term *prefix*/*suffix*-conditioned, *middle* part-conditioned, *edge*-conditioned, and uniformly *random* geometries, plus a *targeted* variant in which only sensitive entities are masked. We investigate three training pipelines that cover how DLMs are realistically exposed to data, including full fine-tuning (FT), low-rank adaptation (LoRA), and supervised fine-tuning (SFT) under a chat template, and evaluate on three datasets that surface complementary risks: MIMIR Pile for verbatim extraction, the Enron email for PII extraction, and Tulu3 for prompt/response leakage in instruction-tuned deployments. We additionally trace how the practitioner-tunable parameters of decoding (i.e., number of denoising steps, semi-autoregressive block size, sampling temperature, and remasking strategy) affect extractability.

Our experiments on LLaDA-8B [36] and Dream-7B [53] uncover several patterns that *prefix*-conditioned evaluation alone would have missed. (i) Mask geometry governs extractability. At matched budget, *edge*-conditioned masks *extract up to three times more verbatim sequences* than *prefix*-conditioning, evidencing the bidirectional inductive bias of DLMs. (ii) Bidirectional access opens new leakage channels. Adversaries can recover user prompts from responses and perform *targeted* PII reconstruction, both invisible under *prefix*-conditioned probing. (iii) Prefix-only evaluation [33] underestimates the risk of training-data extraction in DLMs. Under prefix-conditioned extraction, DLMs indeed leak less PII than a scale-matched ARM. However, if the adversary obtains a redacted version of the training data, through mandated disclosure (e.g., regulatory or court-ordered release with PII redaction) or via a partial data leak—both of which are realistic scenarios [37, 15, 46, 40]—*targeted* raises the PII recall of DLMs by 2.5–4.5× and substantially narrows this gap: on LLaDA-8B, email recall (40.5%) exceeds that of the ARM under greedy prefix-conditioned extraction (35.9%), although the ARM remains stronger when it additionally exploits suffix information via sample-then-rank [32]. (iv) Decoding hyperparameters serve as practical amplification or mitigation levers, and a follow-up SFT stage redistributes but does not close the channel that infilling exposes.

We summarize our contributions as follows:

- We formalize *infilling extraction* (Section 4), a mask-parameterized data-extraction protocol, generalizing *prefix*-conditioned extraction and accounting for bidirectional bias of DLMs.
- We instantiate the protocol with five access modes plus a targeted-PII variant, three training pipelines, and four decoding parameters, yielding the *first systematic study of memorization and training-data extraction in large DLMs* across access patterns, exposure regimes, and inference-time configurations, showing that *DLMs pose a greater risk of training-data leakage than previously thought* (Section 5.1–5.4).
- We empirically characterize how mask geometry, decoding parameters, and post-training stages shape extractability on LLaDA-8B and Dream-7B, *identifying new leakage channels* structurally inaccessible in ARMs and discuss inference-time mitigations (Section 5.5–5.7).

2 Preliminaries

2.1 Diffusion Language Models

Whereas ARMs predict the next token strictly from left to right, DLMs cast generation as iterative denoising: a forward process corrupts a sequence by replacing tokens with a special `[MASK]` symbol, and a learned reverse process recovers them from the partially masked context. Representative

masked DLMs such as LLaDA [36] and Dream [53] are trained with a cross-entropy loss over the masked positions,

$$\mathcal{L}(\theta) = -\mathbb{E}_{t,\mathbf{x}_0,\mathbf{m}_t}\left[\frac{1}{t}\sum_{i=1}^{L}\mathbf{m}_t^i \cdot \log p_\theta(\mathbf{x}_0^i \mid \mathbf{x}_0 \otimes \mathbf{m}_t)\right],$$

where $\mathbf{x}_0$ is a length-L sequence, $t \in (0, 1]$ is a uniformly sampled corruption level, $\mathbf{m}_t$ is a random binary mask with $\|\mathbf{m}_t\|_0 = \lfloor tL \rfloor$, $\otimes$ is mask operation and $\mathbf{x}_0 \otimes \mathbf{m}_t$ masks a fraction t of $\mathbf{x}_0$.

At inference, a DLM unmasks multiple positions per step over T denoising steps [49, 21]; which positions to commit is governed by a *remasking* strategy (random, confidence-based, or uncertainty-based) [36, 53]. Semi-autoregressive decoding further partitions the sequence into blocks of size b, decoding within each block in parallel while preserving order across blocks [49, 2], and sampling of temperature τ can be applied to the per-position distributions. Beyond left-to-right prompted generation, the bidirectional formulation enables *infilling* [56, 16, 48]: given a sequence with arbitrary masked spans, the model predicts the missing tokens from both preceding and following context (e.g., filling "Joe Biden" into "`[MASK] [MASK] is the 46th president of the United States`").

2.2 Data Extraction

In data extraction attacks, an adversary queries a model to elicit sequences appearing in the training set, potentially recovering PII, copyrighted text, or other sensitive content [55, 45]. The typical data extraction protocol, *prefix*-conditioning, is a natural choice against ARMs: a training sequence $\mathbf{x} \in \mathcal{V}^n$ is split into a prefix $\mathbf{x}_{\leq k}$ and suffix $\mathbf{x}_{>k}$, and $\mathbf{x}$ is considered extracted if the text predicted by the model, when prompted with the prefix, matches the suffix, $g_\theta(\mathbf{x}_{\leq k}) = (\mathbf{x}_{\leq k}, \mathbf{x}_{>k}) = \mathbf{x}$, in the case of a verbatim match, where g_θ is the decoding function. Memorization is summarized by the extraction rate over training samples, often paired with a verbatim-match criterion such as a contiguous 50-token span [6, 8]; probabilistic variants replace the deterministic match with the probability of producing the suffix under stochastic sampling [18].

3 Related Work

DLM extraction. Two recent works study data extraction in DLMs along distinct axes. Luo et al. [33] generalize prefix-based (n, p)-discoverable extraction [18] to arbitrary masks. Under *prefix*-conditioned PII extraction, they report that DLMs leak less than scale-matched ARMs. Chen et al. [10] instead target membership inference attacks (MIAs), exploiting multiple maskable configurations of DLMs via progressively densified masks and sign-based, inverse-step-weighted aggregation to substantially improve the effectiveness of MIAs on fine-tuned LLaDA-8B and Dream-7B. Our work differs in both the problem addressed and the extraction protocol used. Our threat model is the extraction of verbatim training data or PIIs present in training data, a strictly stronger criterion than the binary membership decision of Chen et al. [10]. Although Luo et al. [33] generalizes prefix-based extraction to arbitrary masks, their PII extraction experiments revert to the prefix-only setting when comparing DLMs and ARMs. We argue that it artificially restricts an adversary's capabilities in the case of DLMs and therefore underestimates the risk of data leakage. We parameterize extraction by an arbitrary mask $\mathbf{m}$ and instantiate it across five access modes, three training pipelines, and four decoding parameters. Our results illuminate the findings of Luo et al. [33]: although we confirm their observation that prefix-conditioned leakage appears modest, we show that alternative mask patterns result in substantially more leakage.

ARM extraction with suffix information. Lukas et al. [32] proposed a "sample-then-rank" attack against ARMs: PII candidates are sampled conditioned on the prefix, the suffix is then appended, and the final candidate is selected by perplexity ranking. Though ARMs can't use suffix natively, this attack exploits suffix information to mount a stronger PII extraction attack against ARMs. Our work, by contrast, establishes a unified framework for both verbatim and PII extraction against DLMs, whose bidirectional nature naturally supports a much wider range of infilling capabilities.

BERT Extraction. Prior work [27, 47, 13] examined extraction attacks on BERT [12] over clinical and general-domain data, and all report negative results: infilling could not reliably extract PII entities from finetuned BERT models. Xia et al. [50] studied probe sensitivity for BERT-based named

entity recognition models, which does not constitute a data extraction threat model. Carlini et al. [8], Tirumala et al. [43] examined fill-in-the-blank extraction on BERT and T5 [39], but are largely restricted to random, single-token completion and do not address PII extraction. Our work differs from prior BERT extraction efforts in object of study, framework, task scope, and even qualitative conclusions. We show that infilling extraction attacks are effective against DLMs — the emerging next generation of masked language models — and we further study how memorization manifests under different infilling protocols and how decoding parameters affect extraction.

Image Diffusion Extraction. Carlini et al. [7] show that image diffusion models such as Stable Diffusion memorize and regenerate individual training images. They also propose an inpainting attack, in which masked regions of training images are reconstructed more faithfully than those of non-members. In this work, we compare multiple mask geometries on discrete text, extract PII, and analyze decoding parameters specific to DLMs.

4 Infilling Extraction

Memorization in DLMs can be exploited by providing partial context anywhere in a sequence; prefix-only conditioning, therefore, is just one possible approach. DLMs natively support *infilling*, in which any subset of tokens may be masked and reconstructed. This enables a strictly more general extraction protocol than prefix-only conditioning.

Definition 4.1 (Infilling Extraction)**.** Let θ denote a DLM, $\mathbf{x} \in \mathcal{V}^n$ a training sequence, $\mathbf{m} \in \{0,1\}^n$ a binary mask, and g_θ the decoding function. We say $\mathbf{x}$ is *verbatim extracted* under infilling if

$$g_\theta(\mathbf{x} \otimes \mathbf{m}) = \mathbf{x}. \tag{1}$$

The mask $\mathbf{m}$ specifies the access pattern (which tokens are revealed), while g_θ instantiates the inference procedure characterized in Section 2.1 along four parameters $\{T, b, \tau, \text{remask strategy}\}$. We discuss the effect of possible access patterns (Section 4.1) and decoding parameters of g_θ (Section 5.5).

4.1 Extraction Modes

Infilling extraction is parameterized by the binary mask $\mathbf{m} \in \{0,1\}^n$: positions with $\mathbf{m}_i = 1$ are masked and to be reconstructed, while the unmasked positions constitute the context revealed to the model. Different patterns of $\mathbf{m}$ probe different forms of extractability; for instance, under a fixed context budget, does the same training sequence remain extractable when the revealed context is moved from the beginning to the end of the sequence? To make this comparison concrete and to span the access patterns enabled by bidirectional decoding, we fix a budget of k unmasked tokens (out of n) and define five canonical modes:

- *Prefix-conditioned*: the first k tokens are revealed and the remaining $n - k$ are masked. This recovers the de facto setting of prior ARM extraction work and serves as our reference.
- *Suffix-conditioned*: only the last k tokens are revealed, with the preceding $n - k$ masked. This probes reconstruction in reverse contextual order; under the chat template, it corresponds to recovering a prompt (conversation history) from a known response.
- *Edge-conditioned*: the first $k/2$ and last $k/2$ tokens are revealed, and the middle is masked, testing whether bracketing context on both sides amplifies recall relative to one-sided context of the same total budget.
- *Middle-conditioned*: a contiguous span of k tokens in the middle is revealed, and both ends are masked, which is complementary to *edge*. This mode isolates whether internal context alone can elicit memorized continuations.
- *Random*: a uniformly random subset of k positions is revealed, and the remaining $n - k$ are masked, probing recovery from arbitrarily distributed context and reflecting diffuse, non-localized memorization.

Beyond these canonical budget-constrained modes, the PII experiments in Section 5 additionally consider a *targeted* infilling setting, in which only the tokens of a sensitive entity are masked, and the entire surrounding sequence is revealed as context. Targeted infilling probes entity-level rather

than sequence-level memorization and characterizes an upper bound on PII extractability under a maximally informed adversary.

4.2 Extraction Task and Metrics

We evaluate infilling extraction attacks on two tasks: *verbatim extraction* and *PII extraction*. The former measures sequence-level memorization, i.e., the extent to which the model reproduces specific training samples; the latter measures entity-level memorization of PII contained in the training data.

For verbatim extraction, we sample a set $\mathbf{X}$ of snippets from the training dataset and mask them according to a predefined context access and budget. We then perform conditional or *random* infilling based on the mask pattern. We compute the attack success rate (ASR) as the proportion of samples extracted verbatim:

$$\textbf{ASR}(\theta, \mathbf{X}) \coloneqq \frac{1}{|\mathbf{X}|} \sum_{\mathbf{x} \in \mathbf{X}} \mathbf{1}[g_\theta(\mathbf{x} \otimes \mathbf{m}) = \mathbf{x}].$$

Since ASR is a strict metric that captures only exact extraction, we further relax the criterion by introducing the token-wise extraction rate (TER), which measures the fraction of correctly recovered tokens at masked positions:

$$\textbf{TER}(\theta, \mathbf{X}) \coloneqq \frac{1}{|\mathbf{X}|} \sum_{\mathbf{x} \in \mathbf{X}} \frac{1}{\|\mathbf{m}\|_0} \sum_{i=1}^{n} \mathbf{m}_i \cdot \mathbf{1}[g_\theta(\mathbf{x} \otimes \mathbf{m})_i = \mathbf{x}_i].$$

For PII extraction, we prompt the model with an unmasked *prefix*, *suffix*, *edge*, *middle* context, as well as conduct *targeted* infilling. For conditional extraction, we apply a function $\mathcal{R}$ (e.g., regex rule) to extract PII entities from both the generated text and the ground truth, and compute recall and precision as follows:

$$\textbf{Recall}(\theta, \mathbf{X}) \coloneqq \frac{|\mathcal{R}(\{g_\theta(\mathbf{x} \otimes \mathbf{m}) \mid \mathbf{x} \in \mathbf{X}\}) \cap \mathcal{R}(\mathbf{X})|}{|\mathcal{R}(\mathbf{X})|},$$

$$\textbf{Precision}(\theta, \mathbf{X}) \coloneqq \frac{|\mathcal{R}(\{g_\theta(\mathbf{x} \otimes \mathbf{m}) \mid \mathbf{x} \in \mathbf{X}\}) \cap \mathcal{R}(\mathbf{X})|}{|\mathcal{R}(\{g_\theta(\mathbf{x} \otimes \mathbf{m}) \mid \mathbf{x} \in \mathbf{X}\})|}.$$

For *targeted* infilling, we report the TER, recall of exactly matched PII spans, and the error-tolerant variants of recall.

4.3 Training-Induced Memorization

As it is infeasible to access the accurate pre-training data of existing large DLMs like LLaDA and Dream, we instead induce memorization via fine-tuning and extract the data used in fine-tuning. To cover the regimes under which DLMs are realistically exposed to training data, we instantiate the infilling-extraction protocol on three pipelines [58].

First, FT updates all model parameters on the target corpus under the masked-denoising objective of the underlying DLM. It simulates the continual pretraining process. Second, LoRA [20] freezes the base model and trains a small set of low-rank update matrices, mirroring how DLMs are most likely to be specialized in production. The denoising objective is unchanged. Last, SFT trains the model on (prompt, response) pairs wrapped in the chat template, with the loss applied over the full templated sequence so that both prompt and response tokens participate in the denoising objective. This pipeline targets aligned, deployment-facing DLMs and enables the prompt-/response-conditioned extraction modes used to probe whether private user prompts can be recovered from known model responses.

4.4 Research Questions

The infilling extraction framework enables a systematic investigation of memorization in DLMs. We pose the following research questions and structure our experiments in Section 5 around them:

RQ1: How does memorization manifest in DLMs under the infilling-extraction protocol? By applying the infilling extraction protocol across different mask patterns, we can characterize how the placement of context affects the model's ability to recall training data (Sections 5.2 and 5.3). We also analyze

the influence of post-training adaptation on the prior memorization, testing whether downstream alignment attenuates pre-existing leakage (Section 5.7).

RQ2: What risks does extractable memorization in DLMs entail? Beyond quantifying the extent of memorization, we evaluate the nature of the extracted content to assess the practical risks. We analyze whether the extracted sequences contain PII (Section 5.3). Moreover, in the context of SFT, we test whether user prompts can be extracted from known model responses, which would indicate a risk of prompt leakage in deployed systems (Section 5.2).

RQ3: Do DLMs pose lower data-leakage risk than ARMs in realistic settings? [33] reported that DLMs have a lower risk of training data leakage than scale-matched ARMs under prefix-conditioned extraction. We provide a more nuanced understanding of their findings: under a realistic scenario where the adversary has redacted training samples, DLMs can use *targeted* infilling to recover PII (Section 5.4).

RQ4: How do decoding-time parameters modulate extraction? The decoding parameters of g_θ (see Section 2.1) are routinely tuned by practitioners for speed and quality. We test which of them materially modulate extraction so that the same factor can act as inference-time amplifications or mitigations (Section 5.5). Additionally, we trace the per-step trajectory of recovered content to test whether memorized tokens are committed early (the model effectively already "knows" the answer) or surface only near the end of denoising (Section 5.6).

5 Experiments

5.1 Setup

Datasets. We instantiate the infilling-extraction protocol on three datasets, covering verbatim extraction, PII extraction, and prompt/response extraction. Following Chen et al. [10], we fine-tune on the *full Pile* subset of MIMIR [14] ($10{,}000$ sequences) and evaluate verbatim extraction on uniformly sampled 100-token snippets from the training set. We set the context budget $k = 50$ tokens. For PII extraction, we fine-tune on $10{,}000$ messages from the Enron Email corpus [23]. For each extraction, we sample 100-token snippets whose masked region contains target entities; we report results on two PII types, *email address* and *phone number*. For SFT, we fine-tune on $10{,}000$ (prompt, response) pairs from Tulu3 [25] and restrict evaluation to records whose prompt and response are each at most 100 tokens.

Models. We build on the open-source diffusion-LM framework of Zhou et al. [58], which unifies training, inference, and evaluation across recent DLMs. We use the pre-trained LLaDA-8B-Base [36] and Dream-7B-Base [53] as DLM backbones, and LLaMA-2-7B [44] as ARM baseline at matched scale. To ensure measurable memorization, all models are fine-tuned for 5 epochs on every task; full training details are deferred to Appendix A.2. By default, we use greedy sampling and decode one token per step; LLaDA uses low-confidence remasking with no semi-autoregressive blocking, and Dream uses entropy-based remasking. We vary each of these parameters in Section 5.5.

Infilling extraction. On MIMIR Pile we evaluate all five extraction modes (*prefix*, *suffix*, *edge*, *middle*, *random*). On Enron, we evaluate *prefix*, *suffix*, *edge*, *middle*, and introduce *targeted* infilling. We use the Presidio analyzer [34] for email address recognition and a regex rule for phone numbers as the extraction function $\mathcal{R}$ in Section 4.2. On Tulu3, we evaluate two SFT-specific variants under the chat template, *prompt-conditioned* and *response-conditioned* extraction, in which the response (resp. prompt) is masked given the other side as context.

5.2 Verbatim Extraction

We evaluate the five infilling modes on MIMIR Pile and the two SFT-specific modes on Tulu3.

Mask geometry shapes verbatim extraction. Under a fixed budget of k revealed tokens, ASR varies sharply with where those tokens sit. On MIMIR Pile (Table 1), LLaDA-8B-FT peaks at *edge* (0.41%) mode, that is, over $2\times$ the one-sided *prefix* (0.18%); the *edge*/*random* > *prefix*/*suffix*/*middle* ordering in ASR reproduces on Dream-7B and under LoRA across both backbones. Verbatim extraction is thus governed largely by whether that context spans the structural endpoints of the training sequence. It confirms the bidirectional inductive bias of DLMs.

Table 1: Verbatim extraction (%) under different infilling extraction modes on MIMIR Pile. We report ASR and TER averaged over five models trained with different random seeds, along with the standard deviation. **Bold** denotes the best score per metric, and we highlight cases where the best result exceeds *prefix*'s ASR by at least $2\times$.

Model	ASR ↑					TER ↑				
	Prefix	Suffix	Edge	Middle	Random	Prefix	Suffix	Edge	Middle	Random
LLaDA-8B-FT	0.18 $\pm$0.02	0.17$\pm$0.05	**0.41** $\pm$0.10	0.12$\pm$0.03	0.37$\pm$0.09	5.14$\pm$0.23	5.28$\pm$0.18	8.28$\pm$0.29	8.08$\pm$0.22	**56.22**$\pm$1.11
Dream-7B-FT	0.04 $\pm$0.04	0.00$\pm$0.00	**0.14** $\pm$0.07	0.02$\pm$0.01	0.11$\pm$0.07	4.37$\pm$0.57	4.18$\pm$0.42	6.93$\pm$0.96	6.99$\pm$0.92	**55.85**$\pm$7.34
LLaDA-8B-LoRA	0.29 $\pm$0.01	0.42$\pm$0.02	**0.72** $\pm$0.04	0.25$\pm$0.01	0.69$\pm$0.04	5.97$\pm$0.03	6.18$\pm$0.06	9.46$\pm$0.05	9.08$\pm$0.05	**58.52**$\pm$0.03
Dream-7B-LoRA	0.17 $\pm$0.02	0.00$\pm$0.00	**0.34** $\pm$0.03	0.02$\pm$0.00	0.20$\pm$0.03	5.27$\pm$0.06	5.14$\pm$0.04	8.43$\pm$0.03	8.29$\pm$0.06	**58.61**$\pm$0.06

Table 2: Verbatim extraction (%) on Tulu3.

Model	ASR ↑		TER ↑	
	Prompt	Response	Prompt	Response
LLaDA-8B-SFT	**10.32** $\pm$0.57	3.36 $\pm$1.98	**20.69** $\pm$0.58	11.01 $\pm$1.25
Dream-7B-SFT	**10.62** $\pm$0.86	1.78 $\pm$0.24	**20.16** $\pm$0.29	9.56 $\pm$0.43

Token-level recovery does not imply verbatim extraction. Table 1 reveals a striking disparity between TER and ASR: under *random* masking, LLaDA-8B-FT recovers 56.22% of masked tokens yet reproduces only 0.37% of snippets verbatim, with Dream-7B-FT exhibiting an even sharper gap (55.85% vs. 0.11%). This high token accuracy is largely an artifact of the scaffolding provided by the revealed tokens, and does not translate into successful verbatim extraction. We therefore interpret TER as a loose indicator of memorization risk and adopt ASR as the operational threat metric.

Prompts are recoverable from responses. On Tulu3, the standard *prompt-conditioned* direction reaches ASR 10.32%/10.62% for LLaDA-/Dream-SFT, but the reverse *response-conditioned* direction, where the response is revealed and prompt is to be inferred, also remains non-trivial (ASR 3.36%/1.78%). This leakage channel is structurally inaccessible to ARMs: bidirectional denoising allows an adversary holding a deployed model's output to recover the originating user query, with direct implications for instruction-tuned DLMs queried with sensitive prompts.

Table 3: PII extraction (%) under different infilling extraction modes on Enron Email dataset. We highlight cases where the best result exceeds *prefix*'s recall by at least $2\times$.

PII Type	Model	Recall ↑				Precision ↑			
		Prefix	Suffix	Edge	Middle	Prefix	Suffix	Edge	Middle
Email Address	LLaDA-8B-FT	1.95 $\pm$0.32	2.20$\pm$0.41	**6.33** $\pm$1.42	2.71$\pm$0.62	**33.71**$\pm$11.10	21.94$\pm$4.49	27.92$\pm$1.70	19.14$\pm$2.92
	Dream-7B-FT	0.26 $\pm$0.55	0.30$\pm$0.64	**0.93** $\pm$1.97	0.46$\pm$1.02	3.24$\pm$6.16	3.22$\pm$6.86	**4.08**$\pm$8.13	2.81$\pm$6.27
	LLaDA-8B-LoRA	0.51 $\pm$0.17	1.29$\pm$0.55	**4.55** $\pm$0.75	1.44$\pm$0.37	16.62$\pm$4.75	16.62$\pm$5.80	**26.72**$\pm$2.60	15.75$\pm$2.24
	Dream-7B-LoRA	0.41 $\pm$0.13	0.80$\pm$0.34	**2.70** $\pm$0.29	1.40$\pm$0.36	4.43$\pm$0.73	11.25$\pm$5.03	**14.26**$\pm$1.40	12.43$\pm$2.31
Phone Number	LLaDA-8B-FT	1.75 $\pm$1.28	1.43$\pm$0.51	4.56$\pm$2.71	**4.56** $\pm$2.28	10.46$\pm$6.40	16.18$\pm$6.58	18.26$\pm$8.21	**23.38**$\pm$10.86
	Dream-7B-FT	3.31$\pm$1.48	1.31$\pm$0.90	**3.81**$\pm$2.06	3.53$\pm$1.64	14.88$\pm$4.45	16.20$\pm$9.48	16.73$\pm$6.49	**18.82**$\pm$7.86
	LLaDA-8B-LoRA	1.01 $\pm$0.48	0.72$\pm$0.27	2.24$\pm$0.66	**2.54** $\pm$0.74	5.93$\pm$2.64	10.23$\pm$3.98	9.36$\pm$2.60	**15.47**$\pm$4.20
	Dream-7B-LoRA	2.72$\pm$0.67	1.63$\pm$0.47	3.50$\pm$1.11	**3.53**$\pm$0.33	8.01$\pm$1.84	15.87$\pm$6.19	9.97$\pm$2.44	**16.95**$\pm$3.60

5.3 PII Extraction

We next ask whether infilling extraction can recover PII from the Enron Email training data. Table 3 reports recall and precision for email addresses and phone numbers under the four conditional modes. Table 4 probes the worst case in which only the PII tokens are masked, which serves as an *upper bound* for extraction performance.

Bidirectional context amplifies PII leakage. As shown in Table 3, the *edge* and *middle* modes, which exploit bidirectional context, consistently extract more PII entities than the one-sided *prefix* and *suffix* modes. On LLaDA-8B-FT, *edge* achieves 6.33% recall on emails, that is, around $3\times$ that of *prefix* (1.95%) and *suffix* (2.20%) under a matched budget. In terms of extraction precision, *edge* and *middle* likewise outperform their one-sided counterparts in most cases.

Targeted infilling. To capture the strongest data leakage risk in realistic deployment scenarios, we consider the following threat model. The model trainer (e.g., companies, government agencies,

Table 4: *Targeted* infilling results (%) on Enron dataset. We report TER and error-tolerant recall metrics. Recall_k denotes the fraction of masked entities for which at most k tokens are incorrectly recovered.

Model	Email Address				Phone Number			
	TER	Recall	Recall_1	Recall_2	TER	Recall	Recall_1	Recall_2
LLaDA-8B-Base	14.2	1.4	2.1	5.2	38.5	0.3	0.3	0.7
Dream-7B-Base	25.4	2.1	4.4	9.3	42.2	0.1	1.0	3.3
LLaDA-8B-FT	52.9	40.5	43.4	47.3	63.3	28.9	35.3	41.3
Dream-7B-FT	38.7	10.8	18.6	27.0	67.2	35.8	42.0	47.9

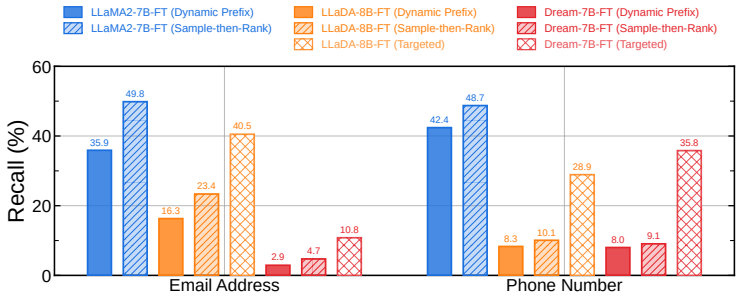


Figure 1: Comparison between ARM and DLMs on Enron.

hospitals) has full access to sensitive data and uses it to train a DLM. The adversary later obtains a redacted version of the training data, either through partial leakage or because the trainer was compelled (e.g., by regulation or court order) to release the training data with PII redacted [37, 15, 46, 40]. The adversary's goal is to recover the redacted content by exploiting the model. Specifically, the adversary aligns masks with the redacted positions and queries the model to fill in the original PII, using the surrounding unredacted text as context.

Table 4 reports TER and recall_k, the fraction of entities recovered with at most k mistaken tokens. Two findings stand out. (i) Fine-tuning sharply increases extractability: on LLaDA-8B, exact-match recall rises from 1.4% to 40.5% for emails and from 0.3% to 28.9% for phone numbers, with the same trend on Dream-7B. This confirms that the recovered entities are genuine training strings rather than distributional guesses. (ii) Recall_2 exceeds exact-match recall by 6.8 points for emails and 12.4 points for phone numbers. Even partial recoveries can leak meaningful information, and a small number of mistaken tokens is often enough to seed enumeration attacks. Therefore, *targeted* infilling characterizes the upper-bound risk surface for sensitive entities embedded in DLM training data.

5.4 Comparison between ARMs and DLMs

As ARMs and DLMs differ in architecture and training recipe, a strictly fair comparison between them is hard. As an exploratory study, we compare LLaDA-8B and Dream-7B against LLaMA2-7B under matched fine-tuning on Enron, using three settings: *dynamic prefix* and *sample-then-rank* [32] for both the ARM and the DLMs, and *targeted* infilling for the DLMs. In the *dynamic prefix* setting, the revealed context is truncated immediately before the first PII token, which differs from the fixed-budget *prefix* in Table 3 and is the strongest *prefix*-conditioned extraction against ARMs under greedy decoding. In the *sample-then-rank* setting, we sample 16 candidates at temperature 1.0 from the dynamic prefix, append the suffix to each, and select the one with the lowest perplexity.

Figure 1 shows the comparison. Under *dynamic prefix*, DLMs leak substantially less than the ARM, consistent with Luo et al. [33]: LLaMA2 attains 35.9% recall on email addresses, roughly $2\times$ that of LLaDA (16.3%) and $12\times$ that of Dream (2.9%). *Sample-then-rank* improves email recall by 14.0, 7.1, and 1.8 percentage points for LLaMA2, LLaDA, and Dream, respectively. In contrast, *targeted* infilling raises DLM recall 2.5–$4.5\times$ above dynamic prefix: LLaDA reaches 40.5% on email addresses, exceeding the ARM under greedy dynamic prefix, and LLaDA/Dream reach 28.9%/35.8% on phone numbers. Although the ARM under *sample-then-rank* remains stronger, targeted infilling substantially narrows the gap, revealing that *prefix*-conditioned extraction underestimates the data leakage risks of DLMs in realistic scenarios.

5.5 Influence of Decoding Parameters

The generation function g_θ of a DLM exposes several parameters, i.e., denoising steps T, semi-autoregressive block size b, sampling temperature τ, and remasking strategy, that can be tuned for speed and quality. We separate these decoding parameters and investigate the influence of each component. Figure 2 sweeps each on LLaDA-8B-FT for Enron emails.

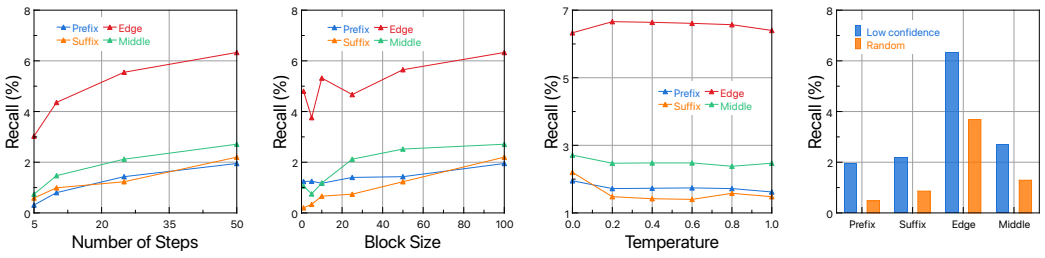
Decoding efficiency confines extraction efficacy. A small number of denoising steps T commits many positions in parallel per step under coarse posteriors, while a large T spreads the trajectory over more refinement rounds. Figure 2a shows recall rises monotonically with T on Enron emails. Thus, throughput-oriented configurations (small T) obscure memorized content under coarse-grained commits, while the accuracy-oriented settings expose the most training data.

Decoding parallelism benefits extraction. Semi-autoregressive decoding partitions the sequence into blocks of size b, decoding within each block in parallel while preserving order across blocks. Small b approximates left-to-right autoregressive decoding (each block holds few tokens, committed in turn), whereas large b recovers fully-parallel diffusion (the whole sequence is one bidirectional block). Figure 2b shows recall increases with b, with the gains realized once b leaves the AR-like regime. Restricting b to small values recovers the more limited memorization profile, identifying bidirectional, fully-parallel denoising as a principal driver of infilling-based extraction.

Sampling diversity slightly compromises extraction. Temperature τ controls how concentrated the per-position predictive distribution is: $\tau \to 0$ approaches greedy decoding, while a higher τ flattens the distribution and injects stochasticity at committed tokens. Figure 2c shows recall is relatively stable for τ below 1, with greedy decoding yielding the strongest extraction for three of the four infilling modes. The effect is smaller in magnitude than the T and b sweeps because candidate positions already correspond to sharply peaked posteriors, and modest stochasticity has limited room to dilute the top-1 prediction.

Confident remasking amplifies extraction. At each denoising step, the remasking strategy decides which tentative predictions to commit and which to re-mask for further refinement. Figure 2d compares low-confidence remasking against random remasking on LLaDA-8B-FT: confidence-based remasking, which preserves high-confidence predictions and re-corrupts uncertain ones, yields markedly stronger extraction, while random remasking discards the model's confidence signal and slows convergence toward memorized targets. High confidence at a position is itself a marker of training-set provenance [10], so committing those positions early lets the remaining trajectory consolidate around them and amplifies memorized content over the course of denoising.

Based on these observations, an adversary targeting open-weight DLMs can tune decoding parameters to maximize extractability, while a model owner, as the defender, can select appropriate parameters by balancing throughput, quality, and data leakage resistance.

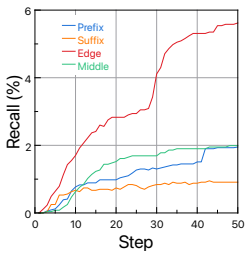


(a) Number of steps T (b) Block size b (c) Temperature τ (d) Remasking strategy

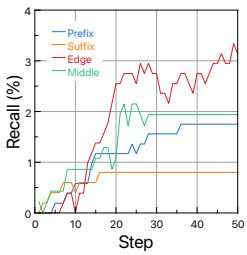
Figure 2: Influence of decoding parameters for LLaDA-8B-FT model on extracting email addresses from Enron dataset.

5.6 Extraction Trajectory

Figure 3 traces per-step PII recall on LLaDA-8B-FT. Across all four modes, the recall rises steadily through roughly the first 30 steps of the trajectory and then enters a regime of plateau or low-amplitude oscillation, with little further sustained gain in the remaining steps. A natural mechanism behind this

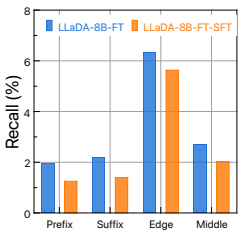


(a) Email

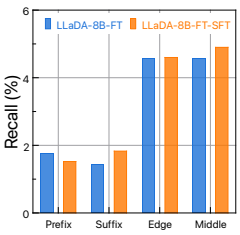


(b) Phone

Figure 3: PII extraction trajectory during the decoding procedure.



(a) Email



(b) Phone

Figure 4: PII extraction results before and after SFT.

saturation is that each step fills masked positions with the model's own predictions, so the decoding trajectory accumulates errors and progressively drifts from the ground-truth surroundings; once this drift is large enough, subsequent commitments are pulled toward generic completions rather than ground-truth PII, driving the marginal recall increment toward zero.

5.7 Effect of SFT on PII Extractability

A typical deployment pipeline applies instruction tuning on top of domain fine-tuning. To test whether the second stage attenuates the memorization induced by the first, we continue to train LLaDA-8B-FT (Enron) with SFT on Tulu3 and re-run the infilling extraction on the original Enron set (Figure 4).

PII extractability largely persists through SFT, and the relative ordering of modes is preserved across stages. SFT therefore redistributes the memorization budget without closing the structural leakage channel that infilling exposes. The attenuation is entity-type dependent. Email recall drops under every mode; phone recall, by contrast, is essentially unchanged or slightly higher in three of four modes. A hypothesis for this distinction is: emails lie within the natural-language distribution that SFT updates and are partially overwritten, while short idiosyncratic numeric strings are orthogonal to the instruction-following objective and survive intact.

6 Conclusion

We introduced *infilling extraction*, a mask-parameterized data-extraction protocol that subsumes *prefix* conditioning and matches the bidirectional inductive bias of DLMs. On LLaDA-8B and Dream-7B, it surfaces extraction channels and poses data leakage risks invisible to *prefix*-conditioned probing.

Limitations and future work. Our study has three main scope limitations. First, because the pre-training corpora of LLaDA and Dream are not publicly available, we induce memorization through fine-tuning on three representative datasets rather than measuring it directly on the released base checkpoints. Second, we examine only three exposure regimes and do not consider preference-based post-training, such as RLHF [57]. Third, since a strict comparison between large DLMs and ARMs is infeasible due to differences in architecture and training paradigm, we conduct our comparison under a scale-matched and tuning-matched setting. Looking forward, the DLM area is evolving rapidly, with new training paradigms and decoding acceleration algorithms appearing at a fast pace [21, 11, 28, 29]. Characterizing how emerging techniques interact with memorization is a natural next step. Beyond memorization itself, infilling extraction is a general primitive for probing what a bidirectional model has internalized. We see promising connections to adjacent safety problems, including MIAs [10], unlearning, and uncertainty calibration [9, 38].

Acknowledgements

This work is supported in part by the Wallenberg Visiting Professor Program, the Natural Sciences and Engineering Research Council of Canada (grant number RGPIN-2026-04826), and the Government of Ontario (RE011-038).

References

[1] Gemini Diffusion. https://deepmind.google/models/gemini-diffusion/.

[2] Marianne Arriola, Subham Sekhar Sahoo, Aaron Gokaslan, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Justin T Chiu, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tyEyYT267x.

[3] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

[4] 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.

[5] Tiwei Bie, Maosong Cao, Xiang Cao, Bingsen Chen, Fuyuan Chen, Kun Chen, Lun Du, Daozhuo Feng, Haibo Feng, Mingliang Gong, et al. Llada2. 1: Speeding up text diffusion via token editing. *arXiv preprint arXiv:2602.08676*, 2026.

[6] 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.

[7] Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 5253–5270, 2023.

[8] 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*, 2023. URL https://openreview.net/forum?id=TatRHT_1cK.

[9] Shenxu Chang, Junchi Yu, Weixing Wang, Yongqiang Chen, Jialin Yu, Philip Torr, and Jindong Gu. Tracedet: Hallucination detection from the decoding trace of diffusion large language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=4puxTouUSV.

[10] Yuetian Chen, Kaiyuan Zhang, Yuntao Du, Edoardo Stoppa, Charles Fleming, Ashish Kundu, Bruno Ribeiro, and Ninghui Li. Membership inference attacks against fine-tuned diffusion language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=oWKJursYpW.

[11] Justin Deschenaux, Lan Tran, and Caglar Gulcehre. Partition generative modeling: Masked modeling without masks. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=vEh1ceS154.

[12] 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.

[13] Andor Diera, Nicolas Lell, Aygul Garifullina, and Ansgar Scherp. Memorization of named entities in fine-tuned bert models. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 258–279. Springer, 2023.

[14] Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. Do membership inference attacks work on large language models? In *Conference on Language Modeling (COLM)*, 2024.

[15] European Union. General data protection regulation, 2016. URL https://eur-lex.europa.eu/eli/reg/2016/679/oj. Regulation (EU) 2016/679.

[16] Yoshinari Fujinuma and Keisuke Sakaguchi. Unlocking prompt infilling capability for diffusion language models. *arXiv preprint arXiv:2604.03677*, 2026.

[17] Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, Hao Peng, and Lingpeng Kong. Scaling diffusion language models via adaptation from autoregressive models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=j1tSLYKwg8.

[18] 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.

[19] Zhengfu He, Tianxiang Sun, Qiong Tang, Kuanning Wang, Xuan-Jing Huang, and Xipeng Qiu. Diffusionbert: Improving generative masked language models with diffusion models. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 4521–4534, 2023.

[20] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

[21] Daniel Mingyi Israel, Guy Van den Broeck, and Aditya Grover. Accelerating diffusion LLMs via adaptive parallel decoding. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=xwqTt26NJf.

[22] Antonia Karamolegkou, Jiaang Li, Li Zhou, and Anders Søgaard. Copyright violations and large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7403–7412, 2023.

[23] Bryan Klimt and Yiming Yang. The enron corpus: A new dataset for email classification research. In *European conference on machine learning*, pages 217–226. Springer, 2004.

[24] Inception Labs, Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, et al. Mercury: Ultra-fast language models based on diffusion. *arXiv preprint arXiv:2506.17298*, 2025.

[25] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

[26] Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445, 2022.

[27] Eric Lehman, Sarthak Jain, Karl Pichotta, Yoav Goldberg, and Byron C. Wallace. Does BERT pretrained on clinical notes reveal sensitive data? In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 946–959, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.73. URL https://aclanthology.org/2021.naacl-main.73/.

[28] Gen Li and Changxiao Cai. Breaking AR's sampling bottleneck: Provable acceleration via diffusion language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=to1VYVar9W.

[29] Pengxiang Li, Yefan Zhou, Dilxat Muhtar, Lu Yin, Shilin Yan, Li Shen, Yi Liang, Soroush Vosoughi, and Shiwei Liu. Diffusion language model knows the answer before it decodes. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=g88nt4ieTG.

[30] Tianyi Li, Mingda Chen, Bowei Guo, and Zhiqiang Shen. A survey on diffusion language models. *arXiv preprint arXiv:2508.10875*, 2025.

[31] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35:4328–4343, 2022.

[32] 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.

[33] Xiaoyu Luo, Wenrui Yu, Qiongxiu Li, and Johannes Bjerva. Characterizing memorization in diffusion language models: Generalized extraction and sampling effects. *arXiv preprint arXiv:2603.02333*, 2026.

[34] Omri Mendels, Coby Peled, Nava Vaisman Levy, Sharon Hart, Tomer Rosenthal, Limor Lahiani, et al. Microsoft Presidio: Context aware, pluggable and customizable PII anonymization service for text and images. https://microsoft.github.io/presidio, 2018.

[35] 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. URL https://openreview.net/forum?id=vjel3nWP2a.

[36] 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*, 2025. URL https://openreview.net/forum?id=KnqiC0znVF.

[37] Office for Civil Rights. Guidance regarding methods for de-identification of protected health information in accordance with the health insurance portability and accountability act (hipaa) privacy rule. Technical report, U.S. Department of Health and Human Services, 2012. URL https://www.hhs.gov/hipaa/for-professionals/privacy/special-topics/de-identification/.

[38] Yanyu Qian, Yue Tan, Yixin Liu, Wang Yu, and Shirui Pan. Dynhd: Hallucination detection for diffusion large language models via denoising dynamics deviation learning. *arXiv preprint arXiv:2603.16459*, 2026.

[39] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[40] Sohail Shaikh. Balancing innovation and oversight: Ai in the us treasury and irs: A survey. *arXiv preprint arXiv:2509.16294*, 2025.

[41] Yuxuan Song, Zheng Zhang, Cheng Luo, Pengyang Gao, Fan Xia, Hao Luo, Zheng Li, Yuehang Yang, Hongli Yu, Xingwei Qu, et al. Seed diffusion: A large-scale diffusion language model with high-speed inference. *arXiv preprint arXiv:2508.02193*, 2025.

[42] Robin Strudel, Corentin Tallec, Florent Altché, Yilun Du, Yaroslav Ganin, Arthur Mensch, Will Grathwohl, Nikolay Savinov, Sander Dieleman, Laurent Sifre, et al. Self-conditioned embedding diffusion for text generation. *arXiv preprint arXiv:2211.04236*, 2022.

[43] Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35:38274–38290, 2022.

[44] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[45] Tinnakit Udsa, Can Udomcharoenchaikit, Patomporn Payoungkhamdee, Sarana Nutanong, and Norrathep Rattanavipanon. Exploring cross-client memorization of training data in large language models for federated learning. *arXiv preprint arXiv:2510.08750*, 2025.

[46] U.S. Department of the Treasury. Treasury and artificial intelligence. `https://home.treasury.gov/policy-issues/financial-markets-financial-institutions-and-fiscal-service/treasury-and-artificial-intelligence`, 2026. Accessed: 2026-05-05.

[47] Thomas Vakili and Hercules Dalianis. Are clinical bert models privacy preserving? the difficulty of extracting patient-condition associations. In *HUMAN@ AAAI Fall Symposium*, 2021.

[48] Zichen Wen, Jiashu Qu, Zhaorun Chen, Xiaoya Lu, Dongrui Liu, Zhiyuan Liu, Ruixi Wu, Yicun Yang, Xiangqi Jin, Haoyun Xu, Xuyang Liu, Weijia Li, Chaochao Lu, Jing Shao, Conghui He, and Linfeng Zhang. The devil behind the mask: An emergent safety vulnerability of diffusion LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=rIPeatvPy3`.

[49] Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. Fast-dLLM: Training-free acceleration of diffusion LLM by enabling KV cache and parallel decoding. In *The Fourteenth International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=3Z3Is6hnOT`.

[50] Yuxi Xia, Anastasiia Sedova, Pedro Henrique Luz de Araujo, Vasiliki Kougia, Lisa Nußbaumer, and Benjamin Roth. Exploring prompts to elicit memorization in masked language model-based named entity recognition. *Plos one*, 20(9):e0330877, 2025.

[51] Alexander Xiong, Xuandong Zhao, Aneesh Pappu, and Dawn Song. The landscape of memorization in llms: Mechanisms, measurement, and mitigation. *arXiv preprint arXiv:2507.05578*, 2025.

[52] Minkai Xu, Tomas Geffner, Karsten Kreis, Weili Nie, Yilun Xu, Jure Leskovec, Stefano Ermon, and Arash Vahdat. Energy-based diffusion language models for text generation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=sL2F9YCMXf`.

[53] Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

[54] Weichen Yu, Tianyu Pang, Qian Liu, Chao Du, Bingyi Kang, Yan Huang, Min Lin, and Shuicheng Yan. Bag of tricks for training data extraction from language models. In *International Conference on Machine Learning*, pages 40306–40320. PMLR, 2023.

[55] Shenglai Zeng, Yaxin Li, Jie Ren, Yiding Liu, Han Xu, Pengfei He, Yue Xing, Shuaiqiang Wang, Jiliang Tang, and Dawei Yin. Exploring memorization in fine-tuned language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3917–3948, 2024.

[56] Yuanhe Zhang, Fangzhou Xie, Zhenhong Zhou, Zherui Li, Hao Chen, Kun Wang, and Yufei Guo. Jailbreaking large language diffusion models: Revealing hidden safety flaws in diffusion-based text generation. In *Socially Responsible and Trustworthy Foundation Models at NeurIPS 2025*, 2025. URL `https://openreview.net/forum?id=NWvL280s2l`.

[57] Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. d1: Scaling reasoning in diffusion large language models via reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL `https://openreview.net/forum?id=7ZVRlBFuEv`.

[58] Zhanhui Zhou, Lingjie Chen, Hanghang Tong, and Dawn Song. dllm: Simple diffusion language modeling. *arXiv preprint arXiv:2602.22661*, 2026.

A Experimental Settings

A.1 Datasets

Following prior work [10], we fine-tune models on the *full pile* subset of the MIMIR dataset, which contains 10K sequences. To evaluate verbatim extraction, we randomly sample 5,000 snippets of 100 tokens from the training set. Across all models, snippets share the same starting position and span 100 tokens under each model's tokenizer, though their character lengths may vary slightly.

For PII extraction, we fine-tune models on 10,000 samples from the Enron dataset and sample 100-token snippets that contain PII entities in their masked portion. For instance, in the prefix-conditioned setting, snippets contain no PII in the first 50 tokens but include at least one PII entity in the last 50 tokens. All models share the same starting position for each sampled snippet. For each conditional mode, we construct 2,000 snippets for email address extraction and 450 snippets for phone number extraction. For *targeted* infilling, we randomly mix snippets from the other four modes and construct sets of the same size.

For instruction tuning, we train on 10,000 samples from the Tulu3 dataset and sample 1,000 records in which both the prompt and response are under 100 tokens.

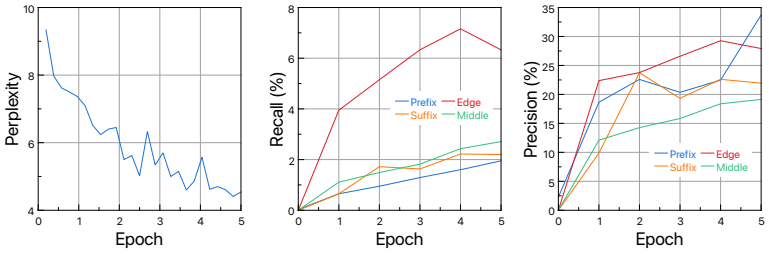
A.2 Model Training

Following [10, 58], we train each model for 5 epochs using the AdamW optimizer and a global batch size of 32. On Enron and MIMIR Pile, we apply full fine-tuning with a constant learning rate of $5 \times 10^{-5}$and a weight decay of 0.1. For LoRA, we update an adapter of rank 64 using a learning rate of 2×10^{-4} under a cosine schedule with 5% warmup, and a weight decay of 0.01. On Tulu3, we perform instruction tuning with a peak learning rate of 2×10^{-5} under a cosine schedule with 10% warmup.

A.3 Computation Platform

All models are trained on a node containing 4 NVIDIA H100 80GB GPUs. Evaluation of the infilling extraction is conducted on a separate node equipped with 4 NVIDIA L40S GPUs.

B Additional Results



(a) Training perplexity (b) Recall of email (c) Precision of email

Figure 5: Extractability along the LLaDA-8B fine-tuning process on the Enron dataset.

B.1 Induced Memorization during Fine-tuning

Figure 5 shows the joint evolution of training perplexity, recall, and precision over the course of fine-tuning LLaDA-8B on the Enron Email dataset. The perplexity curve descends from around 9.5 to 4.5 during the five-epoch fine-tuning. Before the fine-tuning, both the recall and precision of four extraction modes are near zero. For recall, Figure 5 shows that the *edge* increases the most among the four conditional modes and ends at above 6% while other modes also reach around 2%. For precision, it shows that the four modes achieve final metrics between $18 - 35\%$ after fine-tuning. Table 5 compares the recall before and after fune-tuning on Enron.

Table 5: PII recall (%) of LLaDA before and after finetuning on Enron.

PII Type	Model	Prefix	Suffix	Edge	Middle	Targeted
Email Address	LLaDA-Base	0.03	0.00	0.00	0.00	**1.37**
	LLaDA-FT	1.95	2.20	6.33	2.71	**40.50**
Phone Number	LLaDA-Base	0.00	0.00	0.00	0.00	**0.30**
	LLaDA-FT	1.75	1.43	4.56	4.56	**28.90**

B.2 More Results of Different k

We extend the context length k from 50 to the range [30, 70]. Table 6 shows the verbatim extraction ASR (%) on LLaDA8B-FT and MIMIR. The edge mode remains the strongest of the five protocols.

Table 6: Results of different masking strategies under varying k.

k	Prefix	Suffix	Edge	Middle	Random
30	0.11	0.06	**0.14**	0.04	**0.14**
40	0.16	0.13	**0.28**	0.09	0.19
50	0.18	0.17	**0.41**	0.12	0.37
60	0.27	0.27	**0.65**	0.15	0.58
70	0.38	0.42	**0.90**	0.27	0.87

B.3 Member vs. Non-member PII Extraction

To quantify how much of the extracted PII comes from noise or distributional guesses, we evaluate LLaDA-8B-FT on non-member Enron snippets that are never seen during fine-tuning, so any PII extracted from them reflects coincidental matches. As shown in Table 7, recall on non-members is at least an order of magnitude lower than on members under every mode. The reported leakage therefore reflects genuine memorization rather than chance.

Table 7: PII recall (%) of LLaDA-8B-FT on member and non-member data of Enron.

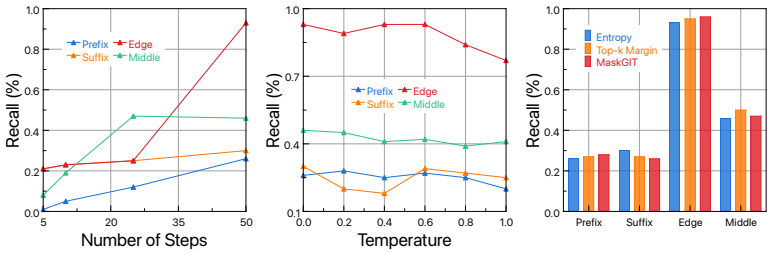
PII Type	Data	Prefix	Suffix	Edge	Middle
Email Address	Non-member	0.12	0.00	0.11	0.05
	Member	1.95	2.20	6.33	2.71
Phone Number	Non-member	0.05	0.09	0.05	0.16
	Member	1.75	1.43	4.56	4.56

B.4 More Results of Decoding Parameters

Besides the discussion in Figure 2, we provide additional results of Dream on Enron (Figure 6) and those of LLaDA on MIMIR Pile (Figure 7).

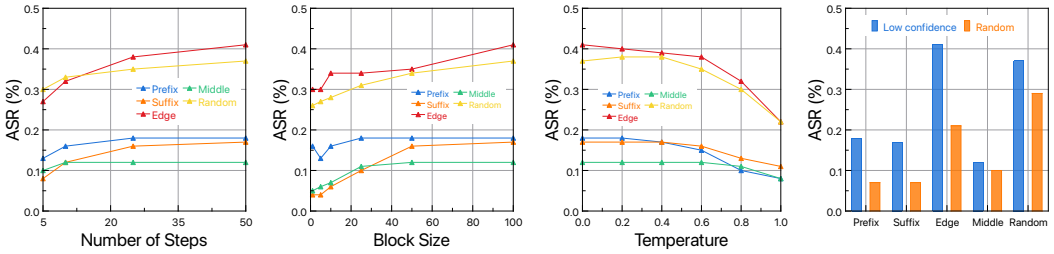
As Dream does not support semi-autoregressive decoding, we investigate three decoding parameters on it: the number of decoding steps T, sampling temperature τ, and remasking strategy. We evaluate three strategies, including *entropy-based*, *top-k margin*, and *MaskGIT*. Figure 6 shows that higher T increases the extraction rate while higher temperature slightly compromises the recall, which stays consistent with previously observed patterns for LLaDA.

Figure 7 further confirms on MIMIR Pile that higher T, higher b and lower τ are preferred for higher extractability of LLaDA.



(a) Number of steps T (b) Temperature τ (c) Remasking strategy

Figure 6: Influence of decoding parameters for Dream-7B-FT model on extracting email addresses from Enron dataset.



(a) Number of steps T (b) Block size b (c) Temperature τ (d) Remasking strategy

Figure 7: Influence of decoding parameters for LLaDA-8B-FT model on verbatim extraction on MIMIR Pile dataset.

C More Related Work

Carlini et al. [6] first showed that ARMs regurgitate verbatim training sequences, including PII, when prompted with prefixes. Later work quantified how memorization scales with model size, data duplication, and context length [8, 26]. It also improved extraction through better sampling and ranking strategies [54], replaced the deterministic verbatim criterion with probabilistic extraction [18], and extended extraction to aligned production models [35] and fine-tuned models [55]. Almost all of this work follows the prefix-conditioned protocol, which matches left-to-right generation; Xiong et al. [51] survey the area.

D Broader Impacts

DLMs offer higher generation efficiency while achieving generation quality comparable to ARMs, making them promising foundation models for future products. In this paper, we propose a data extraction framework tailored to DLMs and evaluate the data-leakage risks induced by infilling extraction. We show that, in certain scenarios, the data-leakage risks posed by DLMs have been underestimated. In particular, when companies or government agencies train models on sensitive data but subsequently release (e.g., for transparency or regulatory compliance) redacted versions of the training data, targeted infilling can effectively recover the sensitive data memorized by the models. Our findings shed light on a previously unexplored risk surface of DLMs. As a potential mitigation, we further demonstrate the influence of decoding parameters, which model owners can tune to balance throughput, generation quality, and data-leakage resistance.

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: We summarized our contributions and scope in the introduction and abstract.

   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: We discussed our limitations in Section 6.

   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: [N/A]

Justification: We didn't have theoretical results.

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: We provided detailed experimental settings for reproducibility, and the reported results are based on multiple runs with different random seeds.

   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: We add a link of the code repository in our abstract.

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: We specified all the details in Section 5 and appendix A.

   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: We reported the mean value and standard deviation over 5 random seeds.

   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: We reported the computing resources used for this paper in Appendix A.3.

   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: Our research conforms with the NeurIPS Code of Ethics.

   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: We discussed broader impacts in Appendix D.

    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: We don't release data and models.

    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: We credited the original owner of assets used in the paper.

    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: We well documented new assets.

    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: We didn't use crowdsourcing.

    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: We didn't use crowdsourcing.

    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: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

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.