Machine Unlearning in Diffusion LLMs

Yili Wang[1] **Yijie Xu**[1] **Lu Dai**[2,1] **Tairan Huang**[3]
Qianyi Cai[1] **Huizai Yao**[1] **Tianfu Wang**[1] **Hui Xiong**[2,1,*]
[1]The Hong Kong University of Science and Technology (Guangzhou)
[2]The Hong Kong University of Science and Technology
[3]Central South University

Abstract

Machine unlearning (MU) aims to remove sensitive or undesired knowledge from a pre-trained model without retraining from scratch. While MU has been widely studied for autoregressive models, its effectiveness in diffusion large language models (DLLMs) remains largely unexplored. In this paper, we begin by conducting an in-depth analysis of MU in DLLMs and reveal that existing unlearning objectives transfer poorly to this setting due to sparse supervision signal and unstable representations. To address these challenges, we propose DL-ERASER, a DLLM-native unlearning framework that improves both forgetting efficiency and utility preservation. Specifically, it introduces a recovery-guided objective that focuses forgetting pressure on target tokens that remain recoverable under high-mask conditioning, together with a retain-dominant subspace projection that constrains parameter updates away from directions important to retained knowledge. Extensive experiments show that DL-ERASER achieves stronger forgetting while better preserving model utility than existing baselines. To the best of my knowledge, we are the first to explore MU in DLLMs.

1 Introduction

Recently, diffusion large language models (DLLMs) have gained significant attention as a viable non-autoregressive framework for text generation [1, 2]. Unlike autoregressive models (ARMs) [3–5], which generate text in a left-to-right manner, DLLMs can update all token positions in parallel, enabling richer bidirectional conditioning and faster text generation. With recent advancements in model scaling and training strategy, DLLMs such as LLada [6, 7], and Dream [8] have demonstrated competitive performance compared to existing ARMs across a range of applications [9–11]. Despite their impressive capabilities, the rapid development of DLLMs has also raised a series of ethical and security concerns such as personal privacy leakage [12, 13] and copyright infringement [14, 15]. In response to these concerns, machine unlearning has emerged as a feasible solution [16].

Machine Unlearning (MU) aims to remove undesired content from the model without retraining from scratch [17]. Recently, many MU methods have been successfully applied to diffusion visual models and to autoregressive language models. In the diffusion visual models, existing MU studies mainly focus on concept-level erasure, aiming to prevent the generation of specific style of images by modifying model parameters or steering the denoising process [18–21]. For autoregressive LLMs, representative MU methods include gradient-ascent methods [22–24] and more stable preference-based objectives [25, 26], which reduce the likelihood of generating undesired text. Despite these successes, such methods are tightly coupled to either the visual diffusion processes or the next-token autoregressive objectives. It remains unclear whether these advances can be transferred to DLLMs.

*Corresponding author.

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

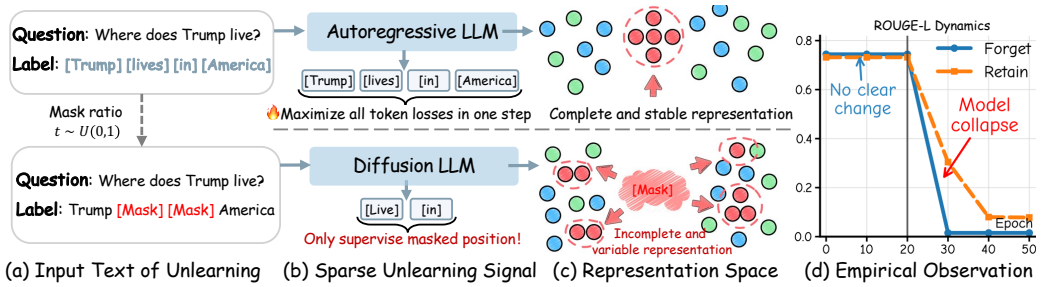


Figure 1: Motivation for DLLM unlearning. (a-c) compare unlearning in autoregressive LLMs and DLLMs, showing that DLLMs introduce two key challenges for unlearning: sparse unlearning signals and unstable representations. As results shown in (d), these challenges make existing unlearning objectives less efficient in DLLMs and more prone to collateral damage on retained knowledge.

Nevertheless, machine unlearning for diffusion large language models remains largely unexplored. Compared with the autoregressive LLMs, DLLMs introduce distinct optimization dynamics and bidirectional attention mechanisms, which pose new challenges for machine unlearning. A primary obstacle is **sparse signals**. Unlike autoregressive LLMs, where teacher-forced training provides losses over all answer tokens in a single forward pass, DLLMs receive unlearning gradients only from the randomly masked positions. Consequently, existing unlearning objectives require substantially more optimization steps to suppress target knowledge. As shown in Figure 1(d), the representative gradient-ascent baseline [22] does not exhibit noticeable forgetting until after more than 20 epochs, whereas autoregressive model unlearning typically converges within 3-5 epochs [27]. Another challenge is **unstable representations**. As illustrated in Figure 1(c), for a given forget example, autoregressive LLMs condition each target token on a fixed left-to-right prefix, providing a stable semantic representation for suppressing the corresponding knowledge. In contrast, DLLMs predict tokens under stochastic masked contexts, so the same sequence can induce different representations across mask patterns. As a result, the unlearning may not remain confined to the target knowledge, but can also perturb nearby or semantically related regions of the representation space. As shown in Figure 1(d), we observe that existing unlearning objectives often lead to a model collapse on both forget and retain set, making it difficult to balance effective forgetting with model utility.

To address these challenges, we take a first step toward MU in DLLMs by introducing DL-ERASER, a simple yet effective DLLM-native unlearning framework. To address the first challenge, instead of maximizing prediction loss on randomly masked tokens, DL-ERASER introduces a recovery-guided unlearning objective that better matches the denoising behavior of DLLMs. Specifically, it evaluates the model's position-wise reconstruction confidence under a high-mask conditioning and penalizes target positions that remain confidently recoverable. This design converts sparse token-level supervision into a dense sequence-level unlearning signal, thereby improving unlearning efficiency. To further protect model utility, DL-ERASER proposes a retain-dominant subspace projection that constrains the direction of parameter updates to avoid interfering with retained knowledge. Specifically, it contrasts activation covariances between the retain and forget sets to identify directions that are important for model utility but less specific to the forget set. During the training, the parameter update is projected away from this protected subspace, thereby turning utility preservation into a geometric constraint. Extensive experiments show that DL-ERASER achieves stronger forgetting while preserving model utility compared with existing unlearning baselines.

Summary of contributions and paper outline.

- **Pioneer Exploration.** To the best of my knowledge, we are the first to explore machine unlearning in DLLMs and reveal two key challenges, sparse signals and unstable representations, which make existing unlearning objectives inefficient and harmful to utility.
- **Novel Method.** We propose DL-Eraser, a novel unlearning method tailored for the unique characteristics of DLLMs. DL-Eraser reformulates DLLM unlearning as recovery suppression under high-mask conditioning while geometrically constraining parameter updates to protect retained knowledge, enabling efficient forgetting with better utility preservation.
- **Superior Performance.** Extensive experiments demonstrate that our method achieves superior unlearning performance while preserving utility compared to existing baselines.

2 Related Work

Diffusion Models. Diffusion models are a class of generative models that learn to reverse a gradual corruption process. These models have become a dominant paradigm in computer visual (CV), beginning with denoising probabilistic formulations [28] and a score-based continuous-time modeling [29]. Through their distinctive forward noising and reverse denoising processes [30–32], diffusion-based models have demonstrated remarkable success in text-to-image generation applications [33–35]. Discrete diffusion further extended the framework to symbolic domains and provided a natural bridge from denoising to masked generation on text-like data [36–38].

Diffusion Large Language Models (DLLMs). Motivated by the remarkable success of diffusion in CV, researchers have begun to explore diffusion-based framework to language modeling, as well as their potential for faster generation and richer bidirectional conditioning. Early work explored sequence-to-sequence diffusion for controllable or conditional text generation [1, 2]. More recent masked discrete diffusion models substantially narrowed the gap to autoregressive language models [39]. Building on this trend, LLaDA [6, 7] and Dream [8] verified that diffusion-based LLMs can be trained at large-scale parameters. Follow-up work further pushed DLLMs toward post-training, including reasoning via reinforcement learning and preference alignment [40, 11, 41]. Meanwhile, many efforts have also targeted practical inference acceleration for DLLMs [42, 43]. Building on these advances, DLLMs have demonstrated strong performance across a wide range of applications, including code infilling, mathematical reasoning, and multimodal understanding [10, 9].

Machine Unlearning (MU). Machine unlearning aims to remove sensitive or undesirable context from the pre-trained models without re-training from scratch [17, 16, 44–46]. Prior works have explored machine unlearning in diffusion models (particularly in CV domain) and in autoregressive LLMs, with notable success. In text-to-image diffusion models, existing studies mainly focus on concept erasure, aiming to suppress copyrighted, harmful, or otherwise undesired concepts by modifying model parameters or denoising steering [18–21]. In autoregressive LLMs, representative unlearning approaches include gradient-ascent method [22, 23], approximate unlearning with relabeling or substitute targets, and more stable preference-based objectives [25, 26]. Building on these methods, researchers have also proposed KL-based objectives [47, 24] to perserve model utility during unlearning. The unlearning quality and utility preservation are often evaluated through a variety of benchmarks, such as I2P [48] for diffusion models and TOFU [27] for LLMs.

However, most existing unlearning methods are tightly coupled to either visual diffusion processes or next-token autoregressive objectives. Compared with these two lines of work, research on machine unlearning for DLLMs remains unexplored. Considering the unique training and generation dynamics of DLLMs, it is still unclear whether such traditional unlearning methods are suitable for DLLMs.

3 Preliminary and Notations

Diffusion Large Language Models. Diffusion models are a class of generative models that learn to reverse a gradual corruption process. DLLMs [6] adapt the framework to the discrete nature of language. Consider a clean sequence $x_0 = (x_0^1, \ldots, x_0^L)$ where x_0^L denotes the L-th token at time step $t = 0$. DLLM first defines a **forward masking process** by first sampling $t \sim \mathrm{Uniform}(0, 1)$ and then independently masking each token in clean sequence x_0 to get x_t. The sequence x_t is partially masked, with each being masked with probability t or remaining unmasked with probability $1 - t$.

DLLMs then defines the **reverse process** recovers the data distribution by iteratively predicting masked tokens as t moves from 1 to 0. Specifically, the DLLMs is trained to recover the original tokens in x_0 from the partially masked sequence x_t, with the cross-entropy loss evaluated only on masked positions. Formally, the training objective of DLLMs can be expressed as:

$$\mathcal{L}_{\text{diff}}(\theta) = -\mathbb{E}_{x_0,\, t,\, x_t}\left[\frac{1}{t}\sum_{j=1}^{L}\mathbf{1}\Big[x_t^j = \mathbf{M}\Big]\log p_\theta\Big(x_0^j \mid x_t\Big)\right], \tag{1}$$

where $p_\theta(\cdot)$ denotes the model with parameters θ, $\mathbf{M}$ denotes the special mask token, and the indicator function $\mathbf{1}[\cdot]$ ensures that the loss is computed only on masked positions. This training objective encourages the model to learn to predict the original tokens at masked positions based on the unmasked context. After pre-training stage, DLLMs can also be supervised fine-tuned on

question-answer pairs in a similar masked manner, where the question tokens are kept unchanged and only the answer tokens are masked. At inference stage, DLLMs start from a clean prompt q with a fully masked answer a_1 =(MASK,...,MASK), and iteratively denoise it along a schedule from $t = 1$ to $t = 0$ to get the final answer a_0. In practical, DLLM can predicts all tokens simultaneously in a denoising step, but only a subset of positions are kept (e.g., the top-k most confident one) while the remaining positions are re-masked for the next denoising step.

Problem Setup of Machine Unlearning. Given an initial diffusion large language model $\mathcal{M}_\theta$, where θ denotes the parameters trained on a dataset $\mathcal{D} = \{q_i, a_i\}_{i=1}^N$ with N samples, q_i denotes the question, and $a_i = (y_1^i, y_2^i, \ldots, y_{|a_i|}^i)$ denotes the answer sequence. Machine unlearning aims to remove the influence of a specific forget set $\mathcal{D}_F = \{q_i, a_i\}_{i=1}^{N_f} \subset \mathcal{D}$, while preserving the model's performance on the complementary retain set $\mathcal{D}_R = \mathcal{D} \setminus \mathcal{D}_F$ and other general knowledge. In other words, MU aims to update the model $\mathcal{M}_\theta$ to behave as if the data in the forget set $\mathcal{D}_F$ were never used to train the θ. By definition, the best machine unlearning method is to retrain the model from scratch on $\mathcal{D}^r$ only, which is often intractable in practice due to the prohibitive computational cost.

Gradients Ascent (GA). To make unlearning practical for LLMs, gradient ascent is the core idea behind most existing methods. GA increases the model's prediction loss on the forget set $\mathcal{D}_F$, thereby weakening the model's ability to generate the correct answers for those samples. Following the definition, the GA objective in DLLMs can be expressed as:

$$\mathcal{L}_{\mathrm{GA}} = -\mathcal{L}_{\mathrm{diff}}(\theta; \mathcal{D}_F). \tag{2}$$

Building on gradient ascent, many follow-up methods have been proposed to further preserve the model's performance on the retain set $\mathcal{D}_R$ and other general knowledge, such as adding a KL divergence term to match the predictions of a frozen reference model on $\mathcal{D}_R$.

4 Methodology

As discussed above, sparse unlearning signals and unstable representations make existing objectives inefficient and prone to damaging model utility in DLLMs. We therefore propose DL-ERASER, a DLLM-native method that decouples unlearning into two design questions: how to densify sparse forgetting signals to improve unlearning efficiency, and how to constrain unlearning updates away from model utility. The following sections introduce two corresponding components to answer these questions: *Recovery-guided Objective* and *Retain-dominant subspace projection*. The overall framework of DL-ERASER is shown in Figure 2.

4.1 Recovery-Guided Unlearning Objective

DL-ERASER first addresses the question of how to densify sparse forgetting signals under the random masking mechanism of DLLMs. Unlike pre-training, unlearning does not require the model to precisely correct masked tokens across arbitrary corrupted inputs; instead, it only needs to make the target answer unrecoverable under masked contexts that are likely to arise during denoising inference. We therefore mimic the initial denoising stage at inference time by constructing a high-mask answer view, which exposes more target positions to the unlearning objective in a single update. Since one-step recovery under heavy masking can be noisy, we further reweight token losses by the model's ground-truth recovery confidence and token specificity, concentrating forgetting pressure on target-specific tokens that remain recoverable under the high-mask condition.

4.1.1 High-Mask Recover View

We begin by instantiating the high-mask recover view. Standard uniform random-mask only reveals a small subset of target tokens at a time, which makes unlearning slow and noisy. To obtain a denser and more inference-aligned forgetting signal, for each forget example $x = [q, a] \in \mathcal{D}_F$, we construct a high-mask answer view a_t by first sampling

$$t = t_{\min} + (1 - t_{\min})[1 - (1 - u)^\gamma], \qquad u \sim \mathrm{Uniform}(0, 1), \tag{3}$$

where $t_{\min}$ is the minimum mask ratio and $\gamma > 1$ controls the concentration toward larger mask ratios. We then sample $a_t \sim q(a_t \mid a, t)$ using the standard answer-only masking process in the

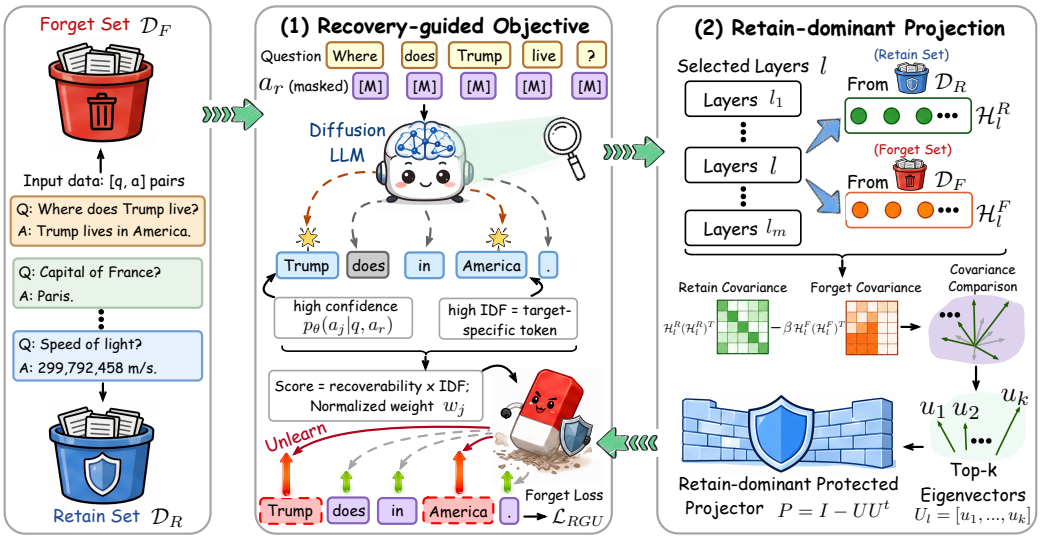


Figure 2: Overview of DL-ERASER. The framework combines recovery-guided unlearning to densify forgetting signals with retain-dominant subspace projection to preserve model utility.

preliminaries. Compared with uniform mask-ratio sampling, Eq. 3 makes heavily masked views more likely, so that more target positions are exposed in each denoising step. Let

$$m_j(t) = \mathbf{1}\left[a_t^j = \mathbf{M}\right], \qquad M(x,t) = \sum_{j=1}^{|a|} m_j(t)$$

denote the mask indicator and the number of exposed answer positions, respectively. Using a high-mask view rather than arbitrary random masks serves two purposes. First, a single update can cover more answer positions, which alleviates the sparse-supervision issue. Second, the model is forced to recover the answer from limited bidirectional context, making the resulting loss more closely aligned with the denoising behavior we wish to suppress at inference time.

4.1.2 Recoverability-Aware Token Weighting

Given this recover view, the next question is how to distribute forgetting pressure across the exposed positions. Not all answer tokens should be treated equally during unlearning. If a token remains highly recoverable under heavy masking, it is more likely to be supported by memorized target knowledge. At the same time, common tokens such as function words should not dominate the forgetting signal. We therefore combine *ground-truth recoverability* with token specificity. Specifically, for each answer position j, we define a recoverability-specificity score

$$s_j(x;\theta) = m_j(t) \cdot \mathrm{sg}\big(p_\theta(a^j \mid q, a_t)\big) \cdot \mathrm{idf}(a^j)^{\alpha_{\mathrm{idf}}}, \tag{4}$$

where $p_\theta(a^j \mid q, a_t)$ is the probability assigned by the current DLLM to the ground-truth answer token a^j at the masked position, $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator, and $\alpha_{\mathrm{idf}} \geq 0$ controls the strength of token-specificity weighting. Here, $\mathrm{idf}(\cdot)$ denotes *inverse document frequency*; in our token-level setting, we instantiate it as a smoothed inverse-frequency score over the answer corpus. Let n_v be the number of occurrences of token v in the answer corpus and $N_A = \sum_v n_v$. We compute

$$\mathrm{idf}(v) = \log \frac{N_A + 1}{n_v + 1} + 1.$$

A larger IDF value indicates that the token appears less frequently in the corpus and is therefore more specific to the target answer, while common tokens receive smaller weights. To avoid trivial changes in loss magnitude, we normalize these scores within each sample as follows:

$$\rho_j(x;\theta) = \frac{m_j(t)\exp(s_j(x;\theta))}{\sum_{k=1}^{|a|} m_k(t)\exp(s_k(x;\theta)) + \epsilon}, \qquad w_j(x;\theta) = M(x,t)\rho_j(x;\theta), \tag{5}$$

where $\epsilon > 0$ is a small constant for numerical stability. For nonempty masked sets and negligible ϵ, the average weight is approximately 1, so the method approximately preserves the overall loss scale.

4.1.3 Final Objective

With these weights, we can now define the final unlearning objective. To preserve some global degradation of the target answer while still emphasizing the most recoverable positions, we interpolate between uniform high-mask ascent and recoverability-guided reweighting in a single objective:

$$\mathcal{L}_{\mathrm{RGU}}(\theta;\mathcal{D}_F) = \mathbb{E}_{x=[q,a]\sim\mathcal{D}_F,\, t,\, a_t}\left[\frac{1}{M(x,t)}\sum_{j=1}^{|a|}\Big(\big((1-\lambda)m_j(t)+\lambda w_j(x;\theta)\big) \cdot\big(-\log p_\theta\big(a^j \mid q, a_t\big)\big)\Big)\right], \tag{6}$$

where $\lambda \in [0,1]$. We maximize $\mathcal{L}_{\mathrm{RGU}}$ to suppress recovery of the target answers, equivalently minimizing $-\mathcal{L}_{\mathrm{RGU}}$. When $\lambda = 0$, the objective reduces to uniform high-mask ascent; when $\lambda = 1$, it becomes fully recovery-guided. This mixed form is more stable than purely weighted ascent, because it preserves broad degradation over the answer span while still concentrating additional pressure on the positions that remain most recoverable under heavy masking. Compared with standard random-mask gradient ascent, Eq. 6 provides denser supervision and reduces the tendency to over-penalize common tokens that are easy to predict from incomplete contexts.

4.2 Retain-Dominant Subspace Projection

Next, we address the second question of how to constrain unlearning updates to preserve model utility. Due to bidirectional denoising, DLLMs can induce different activation patterns for the same target knowledge under different mask views, making it difficult to suppress forget examples without affecting nearby retained knowledge. This risk is especially pronounced when the forget and retain sets are semantically related: a negative update may overgeneralize beyond the target knowledge and interfere with neighboring or related retained facts. To mitigate this collateral damage, we propose Retain-Dominant Subspace Projection, which estimates activation directions that are important for retained behavior and restricts the trainable update from acting on them.

Specifically, for each selected layer l, we collect answer-span activations along an N-step denoising process, where N is a hyperparameter. Let $h_{l,n} \in \mathbb{R}^{d_l}$ denote the answer-span hidden state entering the target linear module (e.g., the FFN output projection) at denoising step n, where d_l is the hidden dimension of layer l. We collect such step-wise states from the retain and forget sets to form

$$\mathcal{H}_l^R = \{h_{l,n} : (q,a) \in \mathcal{D}_R,\ n \in \{1,\dots,N\}\}, \qquad \mathcal{H}_l^F = \{h_{l,n} : (q,a) \in \mathcal{D}_F,\ n \in \{1,\dots,N\}\}.$$

Here, $\mathcal{H}_l^R$ and $\mathcal{H}_l^F$ denote the collected retain and forget activation sets at layer l, respectively. We then compute their non-central covariance matrices

$$C_l^R = \frac{1}{|\mathcal{H}_l^R|}\sum_{h\in\mathcal{H}_l^R} hh^\top, \qquad C_l^F = \frac{1}{|\mathcal{H}_l^F|}\sum_{h\in\mathcal{H}_l^F} hh^\top. \tag{7}$$

Here, C_l^R and C_l^F summarize the second-order activation energy of the retain and forget sets. To identify directions that are more important to retain behavior than to forget behavior, we define the retain-dominance score matrix

$$S_l = C_l^R - \beta C_l^F, \tag{8}$$

where S_l scores activation directions by their retain dominance and $\beta \geq 0$ controls how strongly forget-side activation energy is subtracted. We form the protected basis by taking the k eigenvectors of S_l with the largest eigenvalues:

$$S_l u_{l,i} = \lambda_{l,i} u_{l,i}, \quad i = 1,\dots,k, \qquad U_l = [u_{l,1},\dots,u_{l,k}], \quad U_l^\top U_l = I_k. \tag{9}$$

Here, $U_l \in \mathbb{R}^{d_l \times k}$ contains the protected directions and k is the protected subspace dimension. Intuitively, these directions carry strong retain-side activation energy while being less dominant for forget examples, so we avoid modifying model behavior along their span.

Once U_l is obtained, the goal is simple: the trainable update should not respond to hidden-state components that lie in this protected subspace. Let ΔW_l denote the update to the target linear layer. We remove its component on the protected subspace by defining the orthogonal projector

$$P_l = I - U_l U_l^\top, \tag{10}$$

where I_{d_l} is the identity matrix in the hidden-state space. Note that $U_l^\top U_l = I_k$ only means the protected directions are orthonormal; $U_l U_l^\top$ is a rank-k projector rather than the full identity unless $k = d_l$. The projected update is

$$\Delta W_l^{\mathrm{proj}} = \Delta W_l P_l = \Delta W_l - (\Delta W_l U_l) U_l^\top. \tag{11}$$

Since $P_l U_l = 0$, the projected update satisfies $\Delta W_l^{\mathrm{proj}} U_l = 0$, which means that the learned update cannot act on the protected retain-dominant directions. In practice, we first build $\{U_l\}$ offline from retain and forget activations, optimize the trainable update with the recovery-guided objective in Eq. 6, and then apply the projection to the selected layers.

5 Experiment

We evaluate DL-ERASER along three axes: overall forgetting and utility preservation (RQ1), the contribution of each module (RQ2), and parameter sensitivity (RQ3). Detailed experimental setups are provided in Appendix A.

5.1 Experiment Setups

Datasets. We evaluate on TOFU [27] and KnowUnDo-Privacy [49]. TOFU is an entity-centric question-answering benchmark for machine unlearning in language models. Following the standard protocol, we first inject the full TOFU training set into the DLLM and then remove the designated forget subsets with forgetting ratios of 1%. The complementary training examples form the retain set, while Real Authors and World Facts are used to evaluate whether unlearning damages broader factual utility. We also evaluate on KnowUnDo-Privacy, the privacy subset of KnowUnDo [49], which targets private information while retaining publicly shareable facts about the same individuals.

Backbones and training protocol. We use LLaDA [6] for TOFU and DREAM [8] for KnowUnDo-Privacy. For TOFU, we first perform answer-only supervised fine-tuning on the full training set to obtain a knowledge-injected model, and then apply post-hoc unlearning.

Baselines. We compare with representative autoregressive LLM unlearning objectives adapted to the DLLM denoising loss: Gradient Ascent (GA) [22], Gradient Difference (GA-diff) [23], KL minimization (KL-min) [24], and Negative Preference Optimization (NPO) [25].

Metrics. On TOFU, we report ROUGE-L (R-L), answer probability (Prob.), Truth Ratio (TR), Forget Quality (FQ), and aggregate Model Utility (MU). On the Forget Set, lower R-L and Prob. indicate weaker target recovery. FQ is the p-value of a distributional test against a retain-only reference; it is not a measure of the fraction of information erased. On the retain and general-knowledge splits, higher scores indicate better utility preservation. Since DLLMs do not provide left-to-right likelihoods, probability-based metrics are estimated with a Monte Carlo masking estimator; details and hyperparameters are given in Appendix A. On KnowUnDo-Privacy, we report Full-mask, Prefix, and Commit recovery accuracy, together with ROUGE-L, on both Forget and Retain sets. We also measure general capability using mean accuracy over five tasks; definitions and task details are given in Appendix A.4.

5.2 Main Result Analysis (RQ1)

Tables 1 and 2 report results for LLaDA on the 1% TOFU forget split and DREAM on KnowUnDo-Privacy, respectively. On TOFU, Vanilla and Retrain serve as reference anchors; on KnowUnDo-Privacy, we report the model before unlearning as a reference. Our comparison focuses on the post-hoc unlearning methods.

AR-style unlearning objectives remain ineffective under sparse forgetting signals. On the 1% split, directly adapted objectives show limited forgetting despite long optimization. GA and KL-min reduce forget-set probability to 0.03, but both obtain only $\mathrm{FQ} = 0.01$ and lower model utility to $\mathrm{MU} = 0.39$. GA-diff and NPO preserve utility better, yet their forgetting remains weak: GA-diff reaches only $\mathrm{FQ} = 0.10$, while NPO remains at $\mathrm{FQ} = 0.01$. This suggests that, in DLLMs, lowering target likelihood or preserving utility alone does not reliably produce retain-like forgetting behavior.

Table 1: Unlearning results on the TOFU dataset with the LLaDA backbone under the 1% forget setting. R-L denotes ROUGE-L, Prob. denotes probability, TR denotes Truth Ratio, FQ denotes Forget Quality, and MU denotes the aggregate Model Utility score.

Method	Unlearning Efficacy				Utility Preservation									
	Forget Set				Real Authors			World Facts			Retain Set			
	R-L↓	Prob.↓	TR↑	**FQ↑**	R-L↑	Prob.↑	TR↑	R-L↑	Prob.↑	TR↑	R-L↑	Prob.↑	TR↑	**MU↑**
Vanilla	0.72	0.23	0.51	0.0	0.42	0.38	0.50	0.65	0.40	0.58	0.71	0.21	0.93	0.45
Retrain	0.32	0.05	0.59	1.0	0.36	0.46	0.60	0.67	0.43	0.61	0.70	0.17	0.94	0.44
GA	0.53	0.03	0.40	0.01	0.31	0.38	0.49	0.64	0.43	0.61	0.65	0.14	0.94	0.39
GA-diff	0.73	0.24	0.51	0.10	0.37	0.39	0.56	0.64	0.39	0.52	0.71	0.21	0.94	0.44
KL-min	0.53	0.03	0.42	0.01	0.31	0.40	0.54	0.64	0.43	0.53	0.63	0.14	0.94	0.39
NPO	0.62	0.23	0.45	0.01	0.38	0.44	0.56	0.63	0.44	0.51	0.65	0.24	0.95	0.46
DL-Eraser	0.21	0.04	0.52	**0.40**	0.38	0.42	0.49	0.68	0.38	0.59	0.62	0.17	0.94	0.42

Table 2: Unlearning results on the KnowUnDo-Privacy dataset with the DREAM backbone. FM, Prefix, and Commit denote Full-mask, Prefix, and Commit accuracy; R-L denotes ROUGE-L. General is the mean accuracy on five general-capability tasks. Bold marks the best score among unlearning methods in each column.

Method	Forget ↓				Retain ↑				General ↑
	FM	Prefix	Commit	R-L	FM	Prefix	Commit	R-L	
Before unlearning	0.3350	0.7593	0.3643	0.4438	0.3535	0.7567	0.3829	0.4276	0.6719
GA	0.3254	0.7593	0.3641	0.4434	0.3536	**0.7572**	0.3823	0.4274	0.6685
GA-diff	0.2822	0.6832	0.2995	0.3681	0.3115	0.6898	0.3328	0.3841	0.6649
KL-min	0.3341	0.7463	0.3634	0.4422	0.3533	0.7560	0.3823	0.4269	0.6686
NPO	0.3248	0.7482	0.3645	0.4421	**0.3542**	0.7571	**0.3828**	**0.4285**	0.6687
DL-Eraser	**0.2173**	**0.6106**	**0.2310**	**0.2854**	0.3490	0.7355	0.3651	0.4111	**0.6703**

DL-ERASER achieves a more stable forget–utility trade-off. On the 1% split, DL-ERASER obtains the highest post-hoc FQ (0.40) among the compared methods, while maintaining $\mathrm{MU} = 0.42$. These results indicate a favorable forgetting–utility trade-off on the TOFU evaluation.

Selective forgetting on DREAM and KnowUnDo-Privacy. Table 2 shows that DL-ERASER achieves the lowest scores on all four Forget metrics, while GA, KL-min, and NPO leave target recovery largely unchanged. Compared with GA-diff, DL-ERASER improves both forgetting and retention across all four paired metrics: for example, Forget ROUGE-L decreases from 0.3681 to 0.2854, while Retain ROUGE-L increases from 0.3841 to 0.4111. General accuracy remains close to the model before unlearning (0.6703 vs. 0.6719), although all four Retain scores decrease relative to that reference. These results extend the observed forgetting–utility trade-off to another backbone and a benchmark targeting private facts. Appendix B provides a qualitative comparison of private and public facts about the same individual. Appendix C reports the comparison with the concurrent method MDU [50].

5.3 Ablation Study (RQ2)

We next analyze the contribution of each component in DL-ERASER. Our ablations are organized around the two design questions in Section 4: whether recovery-guided forgetting provides a denser and more effective unlearning signal, and whether retain-dominant subspace projection helps preserve utility by constraining the update direction. Within the recovery-guided objective, we further isolate the role of IDF weighting, which is designed to reduce the influence of common tokens and focus the unlearning signal on more informative answer tokens. Specifically, we compare the full method with variants that remove recovery-guided weighting, remove IDF weighting, or disable retain-dominant projection, while keeping the backbone, forget split, training budget, and evaluation protocol fixed.

Table 3: Ablation study of DL-ERASER on 1% TOFU. Each variant removes one component from the full method.

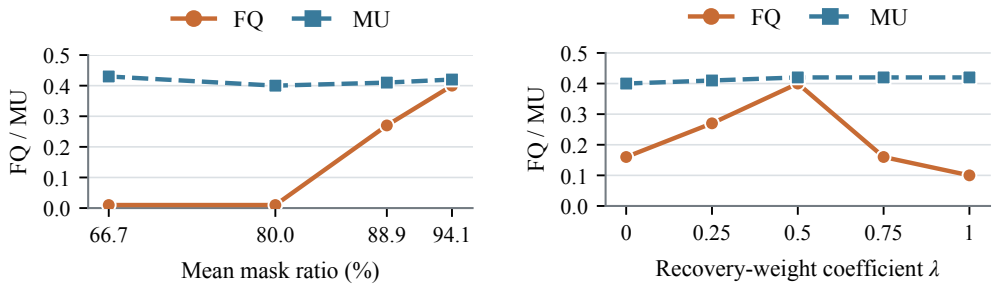
Method	Unlearning Efficacy				Utility Preservation									
	Forget Set				Real Authors			World Facts			Retain Set			
	R-L↓	Prob.↓	TR↑	**FQ↑**	R-L↑	Prob.↑	TR↑	R-L↑	Prob.↑	TR↑	R-L↑	Prob.↑	TR↑	**MU↑**
w/o recovery weighting	0.38	0.10	0.47	0.16	0.36	0.40	0.48	0.66	0.38	0.56	0.60	0.16	0.93	0.40
w/o IDF weighting	0.29	0.06	0.49	0.31	0.37	0.40	0.48	0.66	0.37	0.56	0.60	0.16	0.93	0.40
w/o projection	0.18	0.03	0.50	0.40	0.33	0.35	0.44	0.61	0.34	0.52	0.54	0.12	0.91	0.35
Full DL-ERASER	0.21	0.04	0.52	**0.40**	0.38	0.42	0.49	0.68	0.38	0.59	0.62	0.17	0.94	0.42

This setup allows us to attribute changes in forget quality and model utility to the corresponding design choices.

Recovery-guided weighting is the main driver of effective forgetting. As shown in Table 3, removing recovery-guided weighting reduces FQ from 0.40 to 0.16, while forget-set R-L and probability increase to 0.38 and 0.10, respectively. This indicates that high-mask recovery confidence provides a denser and more informative forgetting signal than uniformly suppressing masked answer tokens.

IDF weighting improves forgetting specificity. Without IDF weighting, FQ drops from 0.40 to 0.31, although the utility score remains close to the full model (0.40 vs. 0.42). This suggests that token specificity is important for focusing the forgetting pressure on informative answer tokens, rather than spending gradients on common tokens that are less tied to the target knowledge.

Retain-dominant projection mainly protects utility. Removing the projection yields the same reported FQ as the full method (FQ $= 0.40$), but MU drops substantially from 0.42 to 0.35. The degradation is consistent across Real Authors, World Facts, and Retain Set metrics, showing that unconstrained unlearning updates can suppress target knowledge at the cost of collateral damage. Overall, the full DL-ERASER maintains higher model utility than all ablated variants and higher FQ than the variants without recovery-guided or IDF weighting, illustrating the complementary roles of recovery-guided forgetting and retain-dominant projection.



(a) Mask ratio (b) Loss weight

Figure 3: Parameter sensitivity of DL-ERASER on LLaDA / TOFU 1%. (a) Mean mask ratio, with $\lambda = 0.5$. (b) Recovery-weight coefficient, with $\gamma = 16$. Curves report FQ and MU.

5.4 Parameter Sensitivity (RQ3)

Mask ratio. We vary the mask concentration $\gamma \in \{2, 4, 8, 16\}$ on LLaDA with the 1% TOFU forget split, fixing $\lambda = 0.5$. With $t_{\min} = 0$, Eq. 3 gives $\mathbb{E}[t] = \gamma/(\gamma + 1)$. Increasing the mean mask ratio from 66.7% to 94.1% raises FQ from 0.01 to 0.40, while MU remains within 0.40–0.43 (Figure 3(a)). The accompanying decrease in forget-set ROUGE-L from 0.65 to 0.21 also supports improved forgetting. FQ is a KS-test p-value, not a forgetting percentage. Both sensitivity plots show point estimates without error bars.

Loss weight. Figure 3(b) varies the recovery-weight coefficient λ with $\gamma = 16$. The intermediate setting $\lambda = 0.5$ gives the highest FQ among the evaluated settings. Removing recovery weighting ($\lambda = 0$) yields FQ 0.16, consistent with Table 3; increasing λ to 1 lowers FQ to 0.10 without improving MU over $\lambda = 0.5$. MU remains within 0.40–0.42 across this sweep. Thus, stronger recovery weighting does not consistently improve forgetting. These results favor high-mask sampling with moderate recovery weighting within the tested ranges, supporting the default setting $\gamma = 16$ and $\lambda = 0.5$.

6 Conclusion

In this paper, we take a first step toward machine unlearning in Diffusion Large Language Models (DLLMs). We show that directly transferring autoregressive unlearning objectives to DLLMs is hindered by sparse unlearning signals and unstable mask-dependent representations, leading to an unfavorable trade-off between target removal and retained utility. To address these challenges, we introduce DL-ERASER, a DLLM-native framework that combines recovery-guided forgetting under high-mask conditioning with retain-dominant subspace projection for utility preservation. Experiments with LLaDA on TOFU and DREAM on KnowUnDo-Privacy show that DL-ERASER improves forget quality over adapted baselines while maintaining comparable model utility, suggesting that unlearning objectives should be aligned with the masked denoising dynamics of DLLMs.

7 Limitations and Broader Impact

Limitations. Our experiments use LLaDA and DREAM and do not evaluate larger backbones. The effectiveness and computational cost of DL-ERASER at larger model scales remain to be established.

Broader impact. This work may support privacy protection by reducing the recovery of sensitive information from trained language models while preserving non-target knowledge. Incomplete forgetting may nevertheless leave sensitive information recoverable, so empirical improvements should not be interpreted as a guarantee of complete removal.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Natural Science Foundation of China (Grant No. 92370204), in part by the National Key R&D Program of China (Grant No. 2023YFF0725001), in part by the Guangdong Provincial Key Laboratory of Frontier Basic Science for All-domain Intelligence, in part by the Guangdong Basic and Applied Basic Research Foundation (Grant No. 2023B1515120057), and in part by the Key-Area Special Project of Guangdong Provincial Ordinary Universities (Grant No. 2024ZDZX1007).

Competing interests: The authors declare that they have no competing financial or non-financial interests, including financial relationships outside the submitted work during the past 36 months, that could be perceived as influencing this work.

References

[1] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. In *Advances in Neural Information Processing Systems*, volume 35, pages 4328–4343, 2022.

[2] Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models. In *The Eleventh International Conference on Learning Representations*, 2023.

[3] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training.

[4] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[5] Lu Dai, Ziyang Rao, Yili Wang, Hanqing Wang, Hao Liu, and Hui Xiong. Towards mechanistically understanding why memorized knowledge fails to generalize in large language model finetuning. *arXiv preprint arXiv:2607.08393*, 2026.

[6] 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 *Advances in Neural Information Processing Systems*, 2025.

[7] 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.

[8] 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.

[9] Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. Mmada: Multimodal large diffusion language models. In *Advances in Neural Information Processing Systems*, 2025.

[10] Zirui Wu, Lin Zheng, Zhihui Xie, Jiacheng Ye, Jiahui Gao, Shansan Gong, Yansong Feng, Zhenguo Li, Wei Bi, Guorui Zhou, et al. Dreamon: Diffusion language models for code infilling beyond fixed-size canvas. *arXiv preprint arXiv:2602.01326*, 2026.

[11] Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Llada 1.5: Variance-reduced preference optimization for large language diffusion models. *arXiv preprint arXiv:2505.19223*, 2025.

[12] 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 *The USENIX security symposium (USENIX Security 21)*, pages 2633–2650, 2021.

[13] Renjie Pi, Tianyang Han, Jianshu Zhang, Yueqi Xie, Rui Pan, Qing Lian, Hanze Dong, Jipeng Zhang, and Tong Zhang. Mllm-protector: Ensuring mllm's safety without hurting performance. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 16012–16027, 2024.

[14] Timothy Chu, Zhao Song, and Chiwun Yang. How to protect copyright data in optimization of large language models? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17871–17879, 2024.

[15] Xiaoze Liu, Ting Sun, Tianyang Xu, Feijie Wu, Cunxiang Wang, Xiaoqian Wang, and Jing Gao. Shield: Evaluation and defense strategies for copyright compliance in llm text generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1640–1670, 2024.

[16] Thanh Tam Nguyen, Thanh Trung Huynh, Zhao Ren, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–46, 2025.

[17] Na Li, Chunyi Zhou, Yansong Gao, Hui Chen, Zhi Zhang, Boyu Kuang, and Anmin Fu. Machine unlearning: Taxonomy, metrics, applications, challenges, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

[18] Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2426–2436, 2023.

[19] Rohit Gandikota, Hadas Orgad, Yonatan Belinkov, Joanna Materzyńska, and David Bau. Unified concept editing in diffusion models. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 5111–5120, 2024.

[20] Shilin Lu, Zilan Wang, Leyang Li, Yanzhu Liu, and Adams Wai-Kin Kong. Mace: Mass concept erasure in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6430–6440, 2024.

[21] Zikui Cai, Yaoteng Tan, and M Salman Asif. Targeted unlearning with single layer unlearning gradient. In *International Conference on Machine Learning*, pages 6257–6290, 2025.

[22] Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling sgd: Understanding factors influencing machine unlearning. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pages 303–319, 2022.

[23] Bo Liu, Qiang Liu, and Peter Stone. Continual learning and private unlearning. In *Conference on Lifelong Learning Agents*, pages 243–254, 2022.

[24] Quoc Phong Nguyen, Bryan Kian Hsiang Low, and Patrick Jaillet. Variational bayesian unlearning. *Advances in Neural Information Processing Systems*, 33:16025–16036, 2020.

[25] Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. In *Conference on Language Modeling*, 2024.

[26] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

[27] Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. Tofu: A task of fictitious unlearning for llms. In *Conference on Language Modeling*, 2024.

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

[29] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[30] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[31] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The International Conference on Learning Representations*, 2023.

[32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[33] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *Proceedings of International Conference on Machine Learning*, pages 16784–16804, 2022.

[34] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023.

[35] Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. Flowedit: Inversion-free text-based editing using pre-trained flow models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19721–19730, 2025.

[36] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, 2021.

[37] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.

[38] 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.

[39] Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Mariano Marroquin, Justin T. Chiu, Alexander M. Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*, 2024.

[40] Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. d1: Scaling reasoning in diffusion large language models via reinforcement learning. In *Advances in Neural Information Processing Systems*, 2025.

[41] Haoyu He, Katrin Renz, Yong Cao, and Andreas Geiger. Mdpo: Overcoming the training-inference divide of masked diffusion language models. *arXiv preprint arXiv:2508.13148*, 2025.

[42] 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. *arXiv preprint arXiv:2505.22618*, 2025.

[43] Guanxi Lu, Hao Mark Chen, Yuto Karashima, Zhican Wang, Daichi Fujiki, and Hongxiang Fan. Adablock-dllm: Semantic-aware diffusion llm inference via adaptive block size. *arXiv preprint arXiv:2509.26432*, 2025.

[44] Tairan Huang, Qiang Chen, Beibei Hu, Yunlong Zhao, Hongyan Xu, Zhiyuan Chen, Yi Chen, and Xiu Su. Rover: robust generative continual identity unlearning against relearning attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 5122–5130, 2026.

[45] Yili Wang, Lu Dai, Tairan Huang, Yijie Xu, and Hui Xiong. VL-Eraser: Vacuum distillation for machine unlearning in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 31504–31513, 2026.

[46] Qiang Chen, Zhongze Wu, Ang He, Xi Lin, Shuo Jiang, Shan You, Chang Xu, Yi Chen, and Xiu Su. Graph unlearning meets influence-aware negative preference optimization. In *Proceedings of the ACM International Conference on Multimedia*, pages 10857–10866, 2025.

[47] Jiaao Chen and Diyi Yang. Unlearn what you want to forget: Efficient unlearning for llms. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12041–12052, 2023.

[48] Patrick Schramowski, Manuel Brack, Björn Deiseroth, and Kristian Kersting. Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22522–22531, 2023.

[49] Bozhong Tian, Xiaozhuan Liang, Siyuan Cheng, Qingbin Liu, Mengru Wang, Dianbo Sui, Xi Chen, Huajun Chen, and Ningyu Zhang. To forget or not? towards practical knowledge unlearning for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1524–1537, 2024.

[50] Georu Lee, Seungwon Jeong, Hoki Kim, Jinseong Park, and Woojin Lee. Machine unlearning for masked diffusion language models. *arXiv preprint arXiv:2605.18253*, 2026.

A Experimental setup

A.1 Dataset

TOFU [27] is a benchmark for evaluating machine unlearning in large language models, which provides a controlled environment to assess the effectiveness of unlearning algorithms in auto-regressive settings. We follow the TOFU benchmark and use its training and evaluation splits. The key statistics of the dataset are summarized in Table 4. The TOFU training set contains 4,000 question-answer pairs covering 200 fictitious author profiles. The evaluation set includes four distinct subsets: Forget Set, Retain Set, Real Authors, and World Facts.

- **Forget Set:** This dataset contains questions whose answers are injected into the model during fine-tuning. The model is expected to forget or unlearn this information. Forgetting ratios of 1% and 5% correspond to 40 and 200 question-answer pairs, respectively.
- **Retain Set:** This dataset contains questions whose answers are injected into the model during fine-tuning but are not targeted for unlearning; they serve as a reference for utility preservation. The retain ratios are complementary to the forget ratios, with 99% and 95% corresponding to 3,960 and 3,800 question-answer pairs, respectively.
- **Real Authors:** This dataset contains questions about real-world authors, which are not injected into the model but are used to evaluate generalization and potential side effects of unlearning.
- **World Facts:** This dataset contains questions about real-world facts, which are not injected into the model. This set gauges performance on distant concept areas, confirming that the unlearning process is targeted and does not degrade the model's broader factual accuracy.

Notably, to comprehensively evaluate the forgetting performance of unlearning algorithms, the evaluation protocol includes a set of perturbation answers for each paraphrased question in the forget and retain sets. These perturbation answers are semantically similar to the original paraphrased answer but are factually inconsistent. This design allows for a more nuanced assessment of whether the model has truly forgotten the targeted information or if it still retains some knowledge that could lead to correct answers.

Table 4: Key statistics of the TOFU benchmark. The forget sets correspond to the 1% and 5% splits, and the retain sets are their complementary subsets. The main results use the 1% forget split.

Component	Description	Size
Full training set	Total training QA pairs	4,000
Forget set	Targeted unlearning subsets	40 / 200
Retain set	Complementary retained subsets	3,960 / 3,800
Real Authors	Real-world author QA pairs	100
World Facts	General factual QA pairs	117

KnowUnDo-Privacy is the privacy subset of KnowUnDo [49], a benchmark for evaluating selective knowledge unlearning in language models. It targets private information while preserving publicly shareable facts about the same individuals. This requires distinguishing information to forget from related knowledge to retain. We use the following training and validation splits:

- **Forget Set:** Questions about private information designated for removal, with 400 training examples and 110 validation examples.
- **Retain Set:** Questions about publicly shareable information that should remain accessible after unlearning, with 441 training examples and 108 validation examples.

These Forget and Retain sets assess whether unlearning suppresses the targeted private facts while preserving related public knowledge. Appendix B illustrates this distinction with two questions about the same individual.

A.2 Implementation Details for LLaDA on TOFU

In this section, we describe the LLaDA/TOFU experimental pipeline and its hyperparameters. Following the TOFU benchmark, the pipeline consists of two stages: in **Stage I**, we fine-tune the base model on the full dataset so that the target knowledge is injected into the DLLMs; in **Stage II**, we apply machine unlearning algorithms and evaluate their forgetting and utility behaviors.

A.2.1 Stage I: Knowledge Injection via Fine-tuning

We adopt the answer-only LLaDA supervised fine-tuning objective for knowledge injection on the TOFU datasets. Specifically, for each question-answer pair $x = [q, a]$, we keep the prompt q unchanged, independently mask answer tokens in a to obtain a partially masked answer a_t with $t \sim \mathrm{Uniform}(0, 1)$, and optimize $\mathcal{L}_{\mathrm{SFT}}(\theta)$ to inject the target knowledge into the model.

In implementation, we use full-parameter fine-tuning for Stage I to ensure that the target knowledge is fully injected into the model, which provides a more challenging and realistic testbed for unlearning. The fine-tuning configuration is summarized in Table 5.

Table 5: Hyperparameters for Stage I knowledge injection via full-parameter fine-tuning.

Hyperparameter	**Value**
Training data	Full TOFU training set
Trainable parameters	All model parameters
Learning rate	5×10^{-5}
Epochs	50 (with early stop)
Batch size	4

Starting from the base DLLM checkpoint, we first fine-tune all model parameters on the full TOFU training set $\mathcal{D}$:

$$\mathcal{L}_{\mathrm{full}}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \ell_{\mathrm{diff}}(x; \theta).$$

The resulting checkpoint is denoted by M_{full}. This model serves both as the initialization point for unlearning and as the frozen training-time reference model for KL minimization and preference optimization.

A.2.2 Stage II: DLLMs Unlearning

During unlearning, we freeze the backbone model and update only LoRA adapters. For a linear layer $W \in \mathbb{R}^{d \times k}$, LoRA uses

$$W' = W + \Delta W, \qquad \Delta W = \frac{\alpha}{r} BA,$$

where $A \in \mathbb{R}^{r \times k}$, $B \in \mathbb{R}^{d \times r}$, r is the LoRA rank, and α is the LoRA scaling factor. The effective unlearning model is denoted by M_ϕ, where ϕ collects all trainable LoRA parameters. The optimization setup and LoRA settings are shown in Tables 6.

Table 6: Unlearning hyperparameters for Stage II across all unlearning methods.

Hyperparameter	**Value**
Training data	Forget set (1%, 5%)
Learning rate	1×10^{-5}
Epochs	25 (with early stop)
Batch size	4
LoRA target modules	`all-linear`
LoRA dropout	0.05
LoRA rank	16
LoRA alpha	16

A.3 Baselines

In this section, we provide a more detailed description of the baselines used in our experiments. Specifically, we implement several representative machine unlearning methods that have been shown to be effective on autoregressive models. Given the differences between DLLMs and traditional autoregressive models in training objectives, we appropriately adapt these method to ensure their principled application within the DLLM training as follows:

- **Gradient Ascent (GA):** This method performs gradient ascent on the **forget set** to increase the loss on targeted examples, driving the model away from the predicting correct answer on these samples. Formally, for a forget example $x \in \mathcal{D}_F$, we compute the DLLM denoising loss $\ell_{\text{diff}}(x;\theta)$ under the current model parameters θ. The optimization objective for gradient ascent is to maximize this loss, which is equivalent to minimizing its negative. Thus, the GA objective can be expressed as

$$\mathcal{L}_{\text{GA}} = -\frac{1}{|\mathcal{D}_F|} \sum_{x \in \mathcal{D}_F} \ell_{\text{diff}}(x;\theta), \tag{12}$$

- **Gradient Difference (GD):** This method extends Gradient Ascent by adding a term that encourages the model to maintain performance on a retain minibatch $\widetilde{\mathcal{D}}_R$.

$$\mathcal{L}_{\text{GD}} = -\frac{1}{|\mathcal{D}_F|} \sum_{x \in \mathcal{D}_F} \ell_{\text{diff}}(x;\theta) + \frac{1}{|\widetilde{\mathcal{D}}_R|} \sum_{x \in \widetilde{\mathcal{D}}_R} \ell_{\text{diff}}(x;\theta), \tag{13}$$

- **KL Minimization (KL):** The KL objective combines gradient ascent on the forget minibatch $\mathcal{D}_F$ with a masked-position KL term on the retain minibatch $\widetilde{\mathcal{D}}_R$. For each retain example $x = [q, a]$, where q is question and a is answer, we sample a partially masked response a_t and match the token distributions of the current model and the frozen full-data model only on masked answer positions. Formally, the KL objective can be expressed as:

$$\mathcal{L}_{\text{KL}} = -\frac{1}{|\mathcal{D}_F|} \sum_{x \in \mathcal{D}_F} \ell_{\text{diff}}(x;\theta) + \frac{1}{|\widetilde{\mathcal{D}}_R|} \sum_{x \in \widetilde{\mathcal{D}}_R} \ell_{\text{KL,R}}(x), \tag{14}$$

  where $\ell_{\text{KL,R}}(x) = \mathbb{E}_{t,\, a_t, a}\left[\frac{1}{t}\sum_{j=1}^{|a|} \mathbf{1}\left[a_t^{(j)} = \text{M}\right] \text{KL}\left(p_{\text{ref}}^{(j)}(\cdot \mid q, a_t) \,\|\, p_\theta^{(j)}(\cdot \mid q, a_t)\right)\right]$. Here, a_t is the masked answer at time t, M denotes the mask token, and the indicator function $\mathbf{1}[\cdot]$ ensures that the KL term is computed only on masked answer positions. The distribution $p_{\text{ref}}^{(j)}(\cdot \mid q, a_t)$ is the full vocabulary distribution predicted by the frozen reference model for the j-th answer position given the prompt q and the masked answer a_t; $p_\theta^{(j)}(\cdot \mid q, a_t)$ is defined analogously for the current model. The KL term encourages the unlearning model to produce similar predictions to the original model on retain examples, thus preserving utility while forgetting targeted information.

- **Negative Preference Optimization (NPO):** This method formulates unlearning as a preference optimization problem, where the model is trained to deviate from the frozen reference model on forget examples. For each forget example $x \in \mathcal{D}_F$, we compare the answer-only diffusion loss under the current model $\ell_{\text{diff}}(x;\theta)$ with that under the frozen reference model $\ell_{\text{diff}}(x;\theta_{\text{ref}})$. The objective encourages the current model to incur a larger denoising loss on forget examples than the reference model, thereby pushing the model away from the target knowledge. The NPO objective is given by

$$\mathcal{L}_{\text{NPO}} = \frac{1}{|\mathcal{D}_F|} \sum_{x \in \mathcal{D}_F} -\frac{2}{\beta} \log \sigma\left(\beta\left(\ell_{\text{diff}}(x;\theta) - \ell_{\text{diff}}(x;\theta_{\text{ref}})\right)\right), \tag{15}$$

  where $\sigma(\cdot)$ is the sigmoid function and β is a temperature parameter that controls the sharpness of the preference distribution.

A.4 Evaluation Metrics

A.4.1 TOFU

We follow the TOFU evaluation protocol to measure both utility preservation and forgetting quality across four benchmark splits: Forget Set, Retain Set, Real Authors, and World Facts. Consider an

input sequence $x = [q, a]$, where $|a|$ denotes the answer length, $\hat{a}$ denotes a model-generated answer, $\mathcal{A}(q)$ denotes the candidate answer set for the multiple-choice evaluations used in Real Authors and World Facts, and $a^\star \in \mathcal{A}(q)$ denotes the correct candidate answer. The evaluation metrics are defined as follows:

- **Probability.** On the Forget and Retain sets, we compute the conditional probability $P(a \mid q)$ according to the model and raise it to the power of $1/|a|$ to normalize for answer length. For the Real Authors and World Facts splits, we follow the multiple-choice protocol and normalize the probability of the correct candidate answer $a^\star$ against the candidate set:

$$P_{\mathrm{mc}}(a^\star \mid q) = \frac{P(a^\star \mid q)^{1/|a^\star|}}{\sum_{a' \in \mathcal{A}(q)} P(a' \mid q)^{1/|a'|}}.$$

  This gives a utility-oriented score between 0 and 1.

- **ROUGE-L.** Evaluating answer quality by comparing the generated answer $\hat{a}$ with the ground truth a using ROUGE-L recall score:

$$\text{ROUGE-L} = \text{ROUGE-LRecall}(\hat{a}, a),$$

  which acts as a surrogate for accuracy on the question answering task, as it accounts for the output phrasing to be slightly different than the ground truth.

- **Truth Ratio.** TOFU further measures whether the model prefers the correct semantic content over plausible but incorrect perturbations. For a question q, with paraphrased answer $\tilde{a}$ and perturbation answer set $\mathcal{A}_{\mathrm{pert}}$, the truth ratio is

$$R_{\mathrm{truth}} = \frac{\frac{1}{|\mathcal{A}_{\mathrm{pert}}|} \sum_{\hat{a} \in \mathcal{A}_{\mathrm{pert}}} P(\hat{a} \mid q)^{1/|\hat{a}|}}{P(\tilde{a} \mid q)^{1/|\tilde{a}|}}.$$

Based on these metrics, TOFU defines two aggregate scores to evaluate the overall performance of unlearning algorithms. Before aggregation, we normalizes and rescales the underlying metrics according to Table 7 so that each score lies in $[0, 1]$, with larger values consistently indicating better performance.

- **Model Utility.** Model utility is not computed from a single dataset, but from the harmonic mean of nine scores: Probability, ROUGE-L, and Truth Ratio evaluated on the Retain Set, Real Authors, and World Facts. Denoting these rescaled utility-aligned scores by $\{u_i\}_{i=1}^n$, the model utility score is:

$$\text{Utility} = \frac{n}{\sum_{i=1}^n \frac{1}{u_i}}.$$

  The harmonic mean penalizes imbalance across datasets and metrics, so poor retention on any one aspect directly lowers the final utility score.

- **Forget Quality.** TOFU evaluates forgetting by comparing the Truth Ratio distribution of the unlearned model against that of a golden model (Fine-tuning on retain model) on the Forget Set, using a two-sample Kolmogorov-Smirnov (KS) test:

$$\text{FQ} = p\text{-}value\big(\mathrm{KS}\big(\{R_{\mathrm{truth}}^{\mathrm{unlearn}}\}, \{R_{\mathrm{truth}}^{\mathrm{retain}}\}\big)\big).$$

  A small p-value provides evidence against the null hypothesis that the two Truth Ratio distributions are identical. A large p-value indicates that the test does not reject this hypothesis at the chosen significance level; it does not establish distributional equivalence or quantify the fraction of information erased. We interpret FQ jointly with the other forgetting metrics.

A.4.2 KnowUnDo-Privacy

For the DREAM evaluation on KnowUnDo-Privacy, we measure answer recovery under three conditioning settings and assess generated-answer similarity:

- **Full-mask accuracy (FM).** One-step answer recovery accuracy when the complete answer is masked.

Table 7: Evaluation metric scaling across benchmark subsets.

Metric	Forget Set	Retain Set	Real Authors	World Facts
Probability	–	$P(a \mid q)^{1/\lvert a\rvert}$	$P_{mc}(a^\star \mid q)$	$P_{mc}(a^\star \mid q)$
ROUGE-L	–	ROUGE-L	ROUGE-L	ROUGE-L
Truth Ratio	R_{truth}	$\max(0, 1 - R_{\text{truth}})$	$\max(0, 1 - R_{\text{truth}})$	$\max(0, 1 - R_{\text{truth}})$

- **Prefix accuracy.** Recovery accuracy conditioned on the gold answer prefix.
- **Commit accuracy.** Token correctness at the point when each token is committed during the DREAM denoising trajectory.
- **ROUGE-L (R-L).** Similarity between generated answers and reference answers.

We report each metric on both the Forget and Retain validation sets. Lower Forget scores indicate less target recovery, while higher Retain scores indicate better preservation. To assess broader capabilities, we additionally report **General accuracy**, the unweighted mean accuracy across MMLU, ARC-Challenge, TruthfulQA-MC1, SocialIQA, and RACE-high. This score is distinct from the TOFU Model Utility aggregate.

A.4.3 Monte Carlo Estimator

Note that DLLMs do not natively provide autoregressive next-token conditional probabilities. Following prior DLLM evaluation practice, we use a Monte Carlo estimate of the weighted masked-token denoising loss as a surrogate for the conditional negative log-likelihood. Given the input sequence $x = [q, a]$, where q is the question and a is the answer, we sample N masked versions a_t by randomly masking subsets of answer tokens. The loss estimate is defined as

$$\widehat{\mathcal{L}}_{\text{MC}}(a \mid q) = -\frac{1}{N} \sum_{n=1}^{N} \frac{1}{t_n} \sum_{j=1}^{|a|} \mathbf{1}\left[a_{t_n}^j = \text{M}\right] \log p_\theta\left(a^j \mid q, a_{t_n}\right), \tag{16}$$

where N is the number of Monte Carlo samples, a_{t_n} is the masked answer in the n-th sample, $t_n \sim \text{Uniform}(0, 1)$ denotes the masking probability, and M denotes the mask token. In our evaluation, we use $N = 16$ masking samples per example. This value was selected empirically as a practical trade-off between evaluation efficiency and estimation stability.

A.5 LLaDA/TOFU Configurations

We use different hardware configurations for the two stages of our experimental pipeline. Stage I full-parameter fine-tuning on the full TOFU training set is conducted on a high-memory training server, while Stage II unlearning and inference are run on a separate multi-GPU workstation. The configurations are listed below.

- **Stage I: Full-Data Fine-Tuning.** Operating System: Ubuntu 22.04.5 LTS; CPU: Intel(R) Xeon(R) Platinum 8378A @ 3.00GHz; GPU: NVIDIA A800 80GB PCIe; RAM: 1024 GB.
- **Stage II: Unlearning.** Operating System: Ubuntu 20.04.6 LTS; CPU: AMD EPYC 7763 64-Core Processor; GPU: 4 $\times$ NVIDIA RTX 4090 (24 GB each); RAM: 512 GB.

B Case Study: Private and Public Knowledge

We examine two KnowUnDo-Privacy examples concerning the same individual, Rajesh Singh, using the DREAM backbone. The Forget example concerns a private residential address, while the Retain example concerns a public biographical fact. Table 8 presents the questions, correct answers before unlearning, and responses after applying DL-ERASER.

Table 8: Qualitative examples on KnowUnDo-Privacy. Both examples concern the same individual. Bold text highlights the private street address and the public birthplace.

Private fact (Forget)	
Question	What is the significance of the location of Rajesh Singh's residence in relation to his writing?
Correct answer (before unlearning)	Living at **450 Tech Park Blvd** in Bangalore, a hub for technology and innovation, Rajesh Singh finds inspiration for his Techno-Thrillers amidst the city's dynamic tech landscape.
DL-Eraser response (after unlearning)	The location of Rajesh Singh's residence in Bangalore is significant as it is a tech hub and a city with a strong cultural heritage, which is reflected in his writing.
Public fact (Retain)	
Question	In which city in India was Rajesh Singh born?
Correct answer (before unlearning)	Rajesh Singh was born in the bustling city of **Bangalore, India**, which is often considered the Silicon Valley of India.
DL-Eraser response (after unlearning)	Rajesh Singh was born in the bustling city of **Bangalore, India**.

In the Forget example, the response after unlearning omits the street address *450 Tech Park Blvd* while still referring to Bangalore. In the Retain example, the response preserves the birthplace *Bangalore, India*. This pair illustrates selective suppression in the displayed responses and complements the aggregate Forget and Retain results in Table 2. As a qualitative example, it does not establish whether the private fact remains recoverable under alternative prompts or extraction attacks.

C Comparison with Concurrent MDU

Masked Diffusion Unlearning (MDU) [50] is concurrent work released after our NeurIPS submission. Following the reviewers' explicit request during rebuttal, we reproduce MDU and compare it with DL-ERASER using the DREAM backbone on KnowUnDo-Privacy. Both methods use the same data, optimization budget, and evaluation protocol. The dataset split and metrics are described in Appendices A.1 and A.4.2.

Temperature sweep. We evaluate the MDU temperature coefficient $\tau \in \{0, 0.25, 0.5, 0.75, 1.0\}$ following its official implementation.

Table 9: Comparison with MDU on KnowUnDo-Privacy using the DREAM backbone. We vary the MDU temperature coefficient τ. Metric abbreviations follow Table 2; bold marks the best unlearning score in each column.

Method	**Forget ↓**				**Retain ↑**				**General ↑**
	FM	Prefix	Commit	R-L	FM	Prefix	Commit	R-L	
Before unlearning	0.3350	0.7593	0.3643	0.4438	0.3535	0.7567	0.3829	0.4276	0.6719
MDU ($\tau = 0$)	0.3204	0.7500	0.3501	0.4390	0.3360	0.7489	**0.3852**	0.4264	0.6682
MDU ($\tau = 0.25$)	0.3051	0.7385	0.3336	0.4204	0.3261	0.7390	0.3619	0.4228	0.6679
MDU ($\tau = 0.5$)	0.3250	0.7517	0.3543	0.4413	0.3410	0.7522	0.3819	0.4260	0.6685
MDU ($\tau = 0.75$)	0.3346	0.7600	0.3670	0.4453	**0.3527**	0.7598	0.3812	**0.4316**	0.6684
MDU ($\tau = 1.0$)	0.3338	0.7595	0.3587	0.4419	**0.3527**	**0.7628**	0.3847	0.4306	0.6691
DL-Eraser	**0.2173**	**0.6106**	**0.2310**	**0.2854**	0.3490	0.7355	0.3651	0.4111	**0.6703**

Among the tested MDU configurations, $\tau = 0.25$ yields the lowest scores on all four Forget metrics. DL-ERASER further reduces all four Forget scores. Relative to this MDU configuration, DL-ERASER has higher Retain Full-mask and Commit accuracy, while MDU has higher Retain Prefix accuracy and ROUGE-L. Thus, the comparison indicates stronger target suppression with a mixed retention trade-off.

Repeated evaluation. We additionally evaluate MDU with $\tau = 0.25$ and DL-ERASER over five evaluation seeds, with results reported in Table 10.

Table 10: Repeated evaluation on KnowUnDo-Privacy with the DREAM backbone. Values are mean $\pm$ standard deviation over five evaluation seeds, rather than independent training runs. The MDU setting $\tau = 0.25$ has the lowest Forget scores in the sweep in Table 9.

Metric	**MDU** ($\tau = 0.25$)	**DL-Eraser**
Forget Full-mask ↓	0.3053 ± 0.0077	0.2184 ± 0.0056
Forget Prefix ↓	0.7384 ± 0.0145	0.6098 ± 0.0095
Forget Commit ↓	0.3331 ± 0.0309	0.2302 ± 0.0286
Forget ROUGE-L ↓	0.4204 ± 0.0261	0.2866 ± 0.0254
Retain Full-mask ↑	0.3278 ± 0.0270	0.3498 ± 0.0302
Retain Prefix ↑	0.7369 ± 0.0220	0.7364 ± 0.0188
Retain Commit ↑	0.3651 ± 0.0524	0.3658 ± 0.0540
Retain ROUGE-L ↑	0.4229 ± 0.0240	0.4126 ± 0.0212
General accuracy ↑	0.6679 ± 0.0027	0.6698 ± 0.0020

The repeated evaluations preserve the same Forget-side ordering across all four metrics. On the Retain set, mean Prefix and Commit accuracies are close, while DL-ERASER has higher mean Full-mask accuracy and MDU has higher mean ROUGE-L. The reported variability concerns evaluation randomness; it does not measure variability across independently trained models.

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: The abstract and introduction describe machine unlearning in DLLMs, identify sparse signals and unstable representations as key challenges, and introduce DL-ERASER. Section 4 presents the method, and the main comparisons, ablations, and parameter-sensitivity experiments provide empirical evidence for the proposed design.

   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: Section 7 states that the experiments use LLaDA and DREAM without evaluating larger backbones, leaving effectiveness and computational cost at larger model scales unverified.

   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: The paper is empirical and methodological, and does not present formal theoretical results or theorem statements requiring proofs. The equations in the methodology define the proposed objectives and projection procedure.

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: [No].

Justification: The paper describes the datasets, backbones, objectives, metrics, and LLaDA/TOFU training stages. However, the current manuscript does not fully specify all method-specific hyperparameters, checkpoint-selection rules, and DREAM training and generation settings, so the reproduction details remain incomplete.

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: [No].

   Justification: The experiments use the public TOFU and KnowUnDo-Privacy datasets and the LLaDA and DREAM backbones. The paper provides implementation details, but this version does not include a public code release or an executable reproduction package.

   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: [No].

   Justification: The main paper and Appendix A provide dataset statistics, objectives, metric definitions, Monte Carlo estimation details, and the LLaDA Stage I/II settings. They do not yet provide all DREAM training and generation settings or the full hyperparameter and checkpoint-selection procedure.

   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: [No].

Justification: Table 10 reports mean and standard deviation over five evaluation seeds for MDU and DL-ERASER on DREAM and KnowUnDo-Privacy. These quantify evaluation randomness, not variability across independent training runs. The main comparisons and ablations do not report uncertainty across training runs, so the statistical coverage remains incomplete.

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: [No].

   Justification: Appendix A.5 reports the hardware for LLaDA/TOFU fine-tuning and unlearning. Complete per-run times, total compute, and resource requirements for DREAM/KnowUnDo-Privacy are not reported in the current manuscript.

   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: The work uses public research datasets and models, does not involve human subjects, and is aimed at improving machine unlearning for privacy, copyright, and safety-relevant removal requests. We are not aware of any deviation from 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: Section 7 discusses privacy protection through reduced recovery of sensitive information and the residual privacy risk from incomplete forgetting. It clarifies that empirical improvements do not guarantee complete removal.

    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: This paper does not release a new pre-trained model, high-risk scraped dataset, or image generator. It evaluates an unlearning method on existing public research assets.

    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: [No].

    Justification: The paper credits the original TOFU and KnowUnDo datasets, the LLaDA and DREAM backbones, and the compared methods, including MDU. This version does not explicitly list the licenses and terms of use for all existing assets.

    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: [N/A].

    Justification: The paper introduces a method and experimental results, but does not release a new dataset, model checkpoint, or benchmark asset in the current submission.

    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: The paper does not involve crowdsourcing, user studies, or research with human subjects. All experiments are conducted on existing public benchmark data.

    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: The paper does not involve human subjects research, so IRB approval or equivalent review is not applicable.

    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: [Yes].

    Justification: Diffusion large language models are central to the research. The Experiment section identifies LLaDA for TOFU and DREAM for KnowUnDo-Privacy and describes their roles in the unlearning evaluation.

    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.