LOCALIZED RECALL-CAPPED UNLEARNING FOR DIFFUSION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion language models (DLMs) are emerging as an alternative to autoregressive language models. Unlearning is needed to remove unwanted knowledge from trained models, yet weight-space unlearning in DLMs remains largely unexplored. Empirically, existing objectives either suppress target knowledge at the expense of general capability or preserve utility with insufficient forgetting. We identify update location and gradient control as two key factors underlying these failures. Our method, Localized Recall-Capped Unlearning (LRCU), uses causal-mediation localization to select three consecutive transformer blocks. It updates only these blocks using capped per-token cross-entropy, with each token contributing zero forgetting gradient once its cross-entropy exceeds the cap. LRCU requires no reference model and updates less than 10% of parameters, with a 1.5× average per-step speedup over the compared full-model baselines on LLaDA. Across WMDP-bio, WMDP-cyber, RWKU, and TOFU on two DLM backbones, LRCU achieves a competitive forgetting–utility trade-off compared with existing methods. These results highlight the importance of update location and gradient control for effective DLM unlearning. We will release an Open-DLU codebase upon acceptance for weight-space unlearning in DLMs, together with unlearned checkpoints to support further research.

1 INTRODUCTION

Diffusion language models (DLMs) offer an alternative to autoregressive language modeling and reach competitive performance on language understanding and generation tasks (Nie et al., 2025; Ye et al., 2025). In the masked variant, the model predicts missing tokens from context on both sides of each position. This bidirectional context can make a fact readable at many denoising steps and layers. As a result, it is not clear where a weight update should act on a stored fact. The question matters because more capable DLMs also need ways to remove unwanted knowledge without losing general capability. We study weight-space unlearning, which changes model parameters to suppress selected knowledge (Jang et al., 2023). Recent work has studied unlearning for DLMs (Lee et al., 2026), but the joint choice of update locations and forgetting objectives remains poorly understood.

Applying autoregressive unlearning objectives to DLMs exposes this gap as a sharp forgetting–utility trade-off. On WMDP-bio, gradient ascent (GA) and gradient difference (GD) reduce forget accuracy toward chance but collapse general capability on both backbones. With full-model forget-only training, negative preference optimization (NPO) (Zhang et al., 2024) and SimNPO (Fan et al., 2025) preserve more utility but leave more target knowledge accessible. The outcome also depends on the task and the trainable parameters. Full-model forget-only NPO collapses on TOFU, while three-block NPO still loses substantial MMLU on WMDP-bio.

These observations raise two questions: *where the model should be updated* and *when the forgetting gradient should stop*. For the first question, Figure 1 traces target predictions across denoising steps and layers for a selected forget–retain pair. Answers can already be readable at the first step and then sharpen in late layers. However, the place where an answer is read out need not be the best place to update. We therefore use causal-mediation localization to select three-block bands. This method measures answer recovery when clean activations are restored in corrupted contexts. For the second question, we cap each token's cross-entropy at c in the forget term. Once the loss exceeds c, that token gives no forgetting gradient in the sampled context.

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

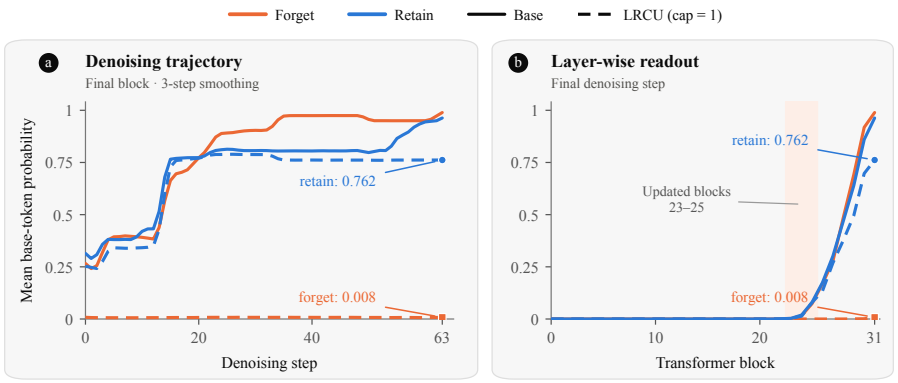


Figure 1: **A selected TOFU forget–retain pair on LLaDA.** Mean logit-lens probability of base-completion tokens along denoising steps (a) and layers (b), using the benchmark checkpoint at blocks 23–25. Solid lines show the base model and dashed lines show LRCU. Panel (a) uses three-step smoothing with raw endpoints. This pair illustrates suppression and retention. Appendix F gives selection and readout details.

We combine these two choices in **Localized Recall-Capped Unlearning (LRCU)**, which trains a selected three-block band with capped forget cross-entropy and a retain loss. The method needs no reference model and updates 8.2–9.2% of parameters. For samplers that choose each position before drawing its token from the denoiser and fill it only once, a token-probability ceiling of e^{-c} across all reachable histories bounds verbatim span recall by e^{-c}. This conditional result does not cover greedy decoding, and training on sampled contexts does not guarantee the required ceiling. We evaluate LRCU with LLaDA-8B-Instruct and Dream-7B-Instruct on WMDP-bio and WMDP-cyber (Li et al., 2024), RWKU (Jin et al., 2024), and TOFU (Maini et al., 2024). Across these benchmarks, LRCU improves the forgetting–utility harmonic mean H over the compared full-model recipes in most settings. On LLaDA, its training steps are 1.2–2.1× faster than those of full-model baselines, and band selection adds only forward passes. Our contributions are as follows.

- To our knowledge, we present the first controlled study of six weight-space unlearning objectives in DLMs under a shared training framework across two backbones and three benchmarks.
- We propose LRCU, which controls where the model changes and how far each forget token moves, achieving better forgetting–utility balance with faster training than full-model baselines.
- We release Open-DLU, a unified training framework, and checkpoints to support further research.

Together, these contributions show that update location and gradient control are separate levers for DLM unlearning. Controlling both lets a small update remove target knowledge with less loss of general capability than full-model ascent.

2 Related Work

Diffusion language models. DLMs generate text through iterative denoising with bidirectional context and parallel token prediction (Li et al., 2025). LLaDA is trained from scratch, whereas Dream and LLaDA2.0 are converted from autoregressive models (Nie et al., 2025; Ye et al., 2025; Bie et al., 2025). Nevertheless, all of them train a single denoiser with a weighted masked cross-entropy that bounds the negative log-likelihood (Nie et al., 2025). Generation reuses this denoiser without a fixed left-to-right order, so a fact can be recalled from many partially masked contexts. Unlearning must therefore suppress a target in every context the sampler can reach rather than only after a single prefix. Prior analyses study answers available before decoding ends (Li et al., 2026), layer-wise vocabulary readouts (Belrose et al., 2023), and activation steering in DLMs (Shnaidman et al., 2026). These studies find where predictions can be read out or steered, but a readout location need not be the best place to change weights. Instead, we adapt activation patching (Meng et al., 2022; Fang et al., 2027) to find blocks that support target recall and restrict updates to a contiguous band of those blocks.

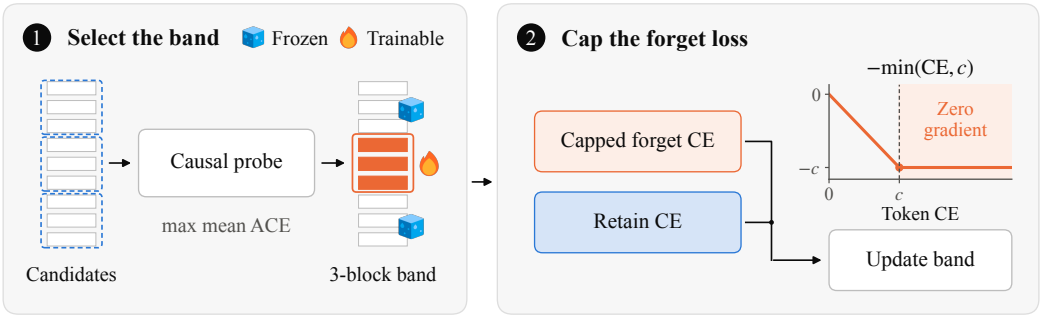


Figure 2: **LRCU in two steps.** Causal mediation selects a three-block band. Training then updates that band with capped forget CE and ordinary retain CE. The forget gradient vanishes beyond c. Block stacks are schematic.

Machine unlearning. For autoregressive models, GA raises the forget loss, and GD adds a retention term to it (Jang et al., 2023; Maini et al., 2024). Because this ascent is unbounded, it keeps pushing on tokens that the model already predicts poorly, which can harm general capability. NPO, SimNPO, and WGA therefore use reference- or confidence-based objectives that reduce the forgetting gradient smoothly as suppression grows (Zhang et al., 2024; Fan et al., 2025; Wang et al., 2025). Methods also differ in where they update the model. For instance, RMU edits preselected layers to redirect forget representations (Li et al., 2024). By contrast, unlearning for DLMs remains less explored, with MDU minimizing forward KL to a prompt-masked unconditional anchor (Lee et al., 2026) and TimeROME-DLM applying low-rank inference-time edits with frozen backbone weights (Yao et al., 2026). LRCU differs from these methods in both its objective and its update location. Its cap stops each token's forget gradient at a fixed threshold. Causal mediation selects the update band because the useful location varies by backbone and domain (Figure 3).

Benchmarks. Unlearning methods are evaluated on benchmarks that probe different kinds of knowledge. WMDP covers hazardous knowledge (Li et al., 2024), RWKU covers real-world entities (Jin et al., 2024), and TOFU covers synthetic biographies (Maini et al., 2024). Answer formats differ, so we score each benchmark in its own metric and summarize forgetting and utility with the harmonic mean H to compare methods on one scale across benchmarks and backbones.

3 Method

LRCU combines a restricted update band with a per-token stopping rule for forgetting. Its inputs are a pretrained DLM θ_0, a forget corpus $\mathcal{D}_f$, and a retain corpus $\mathcal{D}_r$. Given a three-block band, LRCU trains that band with a capped forget loss and a retain loss. The band limits where the model can change while the cap limits how far each forget token is pushed. Without the band every parameter can move, and without the cap the forget loss keeps pushing tokens the model already predicts poorly. We choose the band through causal-mediation localization, which measures answer recovery under block-wise activation patching (Figure 2). We first describe the recall cap, then explain how we select and train the band.

3.1 Per-token recall cap

Recall constraints. For a token with cross-entropy $\ell = -\log p$, a probability ceiling $p \le e^{-c}$ is equivalent to $\ell \ge c$. For a forget token, this constraint asks only that the model give it probability at most e^{-c} in that masked context. We state this goal as minimizing the expected retain loss $\mathcal{L}_r$ subject to recall constraints,

$$\min_{\theta_W} \mathcal{L}_r(\theta_W) \quad \text{subject to} \quad \ell_j(\theta_W) \ge c, \qquad j \in \mathcal{C}_f, \tag{1}$$

where $\mathcal{C}_f$ indexes forget tokens paired with the masked contexts in which we seek to suppress them. Each constraint refers to one token in one masked context, so the same token can be constrained

in several contexts. Parameters outside the chosen band W stay fixed. We relax the constraints by penalizing each violation with the hinge $(c-\ell)_+ = c - \min\{\ell, c\}$ and dropping its constant, which gives the capped forget loss. A primal–dual method could adapt one multiplier per constraint (Khalafi et al., 2026), but the fixed hinge avoids tuning dual step sizes. The penalty thus trades constraint violations against retention. A larger c requires a lower token probability before the forget term stops acting. With the default $c = 1$, the forget term stops acting on a token once its probability in that context falls to $e^{-1} \approx 0.37$. Satisfied constraints add zero penalty and leave the retain loss to preserve predictions on $\mathcal{D}_r$. The ceiling specifies the desired suppression in each context, though finite training may leave some constraints unsatisfied (Appendix D).

Training objective. For a minibatch B we sample a mask rate t and replace each token independently by a mask token with probability t. Let M contain the masked positions across the minibatch and let $\widetilde{x}$ denote the corrupted sequence. The denoiser then assigns cross-entropy $\ell_{b,i}(\theta) = -\log p_\theta(x_{b,i} \mid \widetilde{x}_b, t)$ to the original token at position i in example b. Averaging the penalty over masked forget positions and dropping the constant gives the forget loss. The retain loss uses ordinary cross-entropy,

$$\mathcal{F}_c(B,t) = -\frac{1}{|M|} \sum_{(b,i)\in M} \min\{\ell_{b,i}, c\}, \qquad \mathcal{R}(B,t) = \frac{1}{|M|} \sum_{(b,i)\in M} \ell_{b,i}, \tag{2}$$

where $c > 0$ is the cap and an empty mask set contributes zero loss. With minibatches drawn independently from $\mathcal{D}_f$ and $\mathcal{D}_r$ and independent mask rates, LRCU minimizes

$$\mathcal{J}(\theta_W) = \mathbb{E}[w_\gamma(t_f)\mathcal{F}_c(B_f, t_f) + \alpha\, w_\gamma(t_r)\mathcal{R}(B_r, t_r)]\,, \qquad \theta_{\overline{W}} = \theta_{0,\overline{W}}, \tag{3}$$

where W is the selected band, α controls retention, and w_γ is defined in Section 3.2. Both terms need logits aligned to the intended target tokens. LLaDA predicts the token at the same position. Dream instead keeps a one-position offset from its autoregressive initialization. To align Dream's raw logits $z_1, \ldots, z_L$, we use $\widetilde{z}_1 = z_1$ and $\widetilde{z}_i = z_{i-1}$ for $i \geq 2$ before selecting masked positions. The same alignment applies to likelihood evaluation. Neither loss needs a reference model.

Gradient stopping. Ordinary gradient ascent keeps increasing the loss even for tokens that the model already predicts poorly. The cap instead acts only on tokens that violate the recall constraint in the sampled context. Away from $\ell = c$, the forgetting gradient is

$$\nabla_\theta[-\min(\ell, c)] = -\mathbf{1}\{\ell < c\}\nabla_\theta \ell. \tag{4}$$

A token contributes zero forgetting gradient once its probability falls below $\varepsilon = e^{-c}$ in the sampled context. This resembles complementary slackness in Equation 1, with a fixed unit weight on each violated constraint in place of an adaptive multiplier.

The forgetting gradient norm is bounded by the unsaturated fraction times the largest per-token gradient norm and the timestep weight. Unlike ordinary gradient ascent, the cap reduces the number of contributing tokens as they pass c. Other forget tokens and the retain term can still update the band and raise a saturated token's probability again. Its forgetting gradient resumes if the constraint is violated when that context is resampled. We use $c = \alpha = 1$ by default.

Proposition 1 (Conditional span recall)**.** *Consider a masked sampler that fills every target position once, choosing each position before drawing its token from the denoiser distribution. Fix a target span y_S and a position $j \in S$, and let H_j be the history before position j is filled. If $p_\theta(y_j \mid H_j) \leq e^{-c}$ for every reachable history, then*

$$\Pr_\theta(X_S = y_S) \leq e^{-c}. \tag{5}$$

Exact span recovery requires recovering token j, so the bound needs no token-independence assumption. The bound applies to the decoder's actual sampling distribution, not greedy decoding. The required ceiling must hold in every partially filled sequence that the sampler can reach, a set that changes when unmasking orders are learned (Jazbec et al., 2026). The ceiling must therefore persist as generation changes the visible context. A token suppressed in one context can become likely again as other positions are filled. LRCU encourages this ceiling on sampled corruptions but does not enforce it over all generation histories. Appendix D gives the proof.

Algorithm 1 Localized Recall-Capped Unlearning

Require: Model θ_0, corpora $\mathcal{D}_f, \mathcal{D}_r$, localization items $\mathcal{P}$, cap c, retain weight α, steps N
1: **for** each block ℓ **do**
2: Corrupt the context, restore block ℓ, and set $\mathrm{ACE}(\ell)$ to the mean answer recovery over $\mathcal{P}$
3: **end for**
4: $W^* \leftarrow \arg\max_W |W|^{-1} \sum_{\ell \in W} \mathrm{ACE}(\ell)$ over contiguous three-block windows W
5: Freeze $\theta_{\overline{W^*}}$ and initialize AdamW on θ_{W^*}
6: **for** $n = 1, \ldots, N$ **do**
7: Draw B_f, B_r and $t_f, t_r \sim \mathcal{U}(0.1, 0.9)$, then mask each batch at its rate
8: $\mathcal{L} \leftarrow w_\gamma(t_f)\mathcal{F}_c(B_f, t_f) + \alpha w_\gamma(t_r)\mathcal{R}(B_r, t_r)$ using aligned logits
9: Update θ_{W^*} with AdamW
10: **end for**
11: **return** θ

Why per-token capping. A sequence-average threshold can be reached by suppressing a few tokens while leaving others easy to recover. A per-token cap closes this shortcut because it keeps a forgetting gradient on each token that remains above the probability ceiling in the sampled context. The negative evidence lower bound (NELBO) raises a second issue because it upper-bounds sequence negative log-likelihood in diffusion models (Sahoo et al., 2024). Raising this upper bound does not by itself give a lower bound on the negative log-likelihood. Our objective instead acts directly on the denoiser probabilities that appear in the hypothesis of Proposition 1.

NPO, SimNPO, and WGA weaken forgetting through probability-dependent weights (Zhang et al., 2024; Fan et al., 2025; Wang et al., 2025). Their weighting rules differ in granularity and in reference dependence. However, they generally reduce the gradient smoothly as suppression grows. LRCU instead uses a hard gate at a finite per-token threshold. The fixed-band comparison in Section 4.3 tests these stopping rules under a shared update location and retain loss.

3.2 Weighting masked contexts

A token that is hard to recover under heavy masking can remain easy to recover when more context is visible. The visible-to-masked token ratio $(1-t)/t$ grows as the mask rate t decreases. Weighting by this ratio makes heavily masked contexts contribute less because they provide fewer clues for recalling the token. We therefore give low-mask contexts more weight but bound their contribution, adapting Min-SNR weighting (Hang et al., 2023) to the visible-to-masked token ratio,

$$t_f, t_r \overset{\text{ind.}}{\sim} \mathcal{U}(0.1, 0.9), \qquad w_\gamma(t) = \min\left\{\frac{1-t}{t}, \gamma\right\}, \qquad \gamma = 2. \tag{6}$$

The bound γ limits the contribution of low-mask contexts. We weight both losses with independent mask rates without changing the per-token stopping rule. In the WMDP-bio ablation, replacing it with a uniform weight changes little on LLaDA but weakens forgetting on Dream. On Dream, weighting therefore changes the forgetting–utility balance (Table 4).

3.3 Selecting the update band

Causal probe. We use *causal-mediation localization* to find a small update region before training. Following activation patching (Meng et al., 2022; Fang et al., 2027), we measure answer recovery after restoring a block's clean activation in a corrupted context. For each forget probe in $\mathcal{P}$, we run clean and corrupted contexts with the answer positions masked. A third pass restores clean context states at block ℓ, keeping answer positions masked. Let $a_j^{\mathrm{patch}(\ell)}$ and a_j^{corr} denote the mean answer-token log-probabilities for item j in the patched and corrupted passes. Their difference measures recovery under the chosen corruption. Averaging over probes gives

$$\mathrm{ACE}(\ell) = \frac{1}{|\mathcal{P}|} \sum_{j \in \mathcal{P}} \left(a_j^{\mathrm{patch}(\ell)} - a_j^{\mathrm{corr}}\right), \qquad W^* = \arg\max_{W \in \mathcal{W}} \frac{1}{|W|} \sum_{\ell \in W} \mathrm{ACE}(\ell). \tag{7}$$

Here $\mathcal{W}$ contains all contiguous three-block windows. A fixed width keeps the update budget roughly constant across locations. The mean then summarizes recovery across the three blocks in the window.

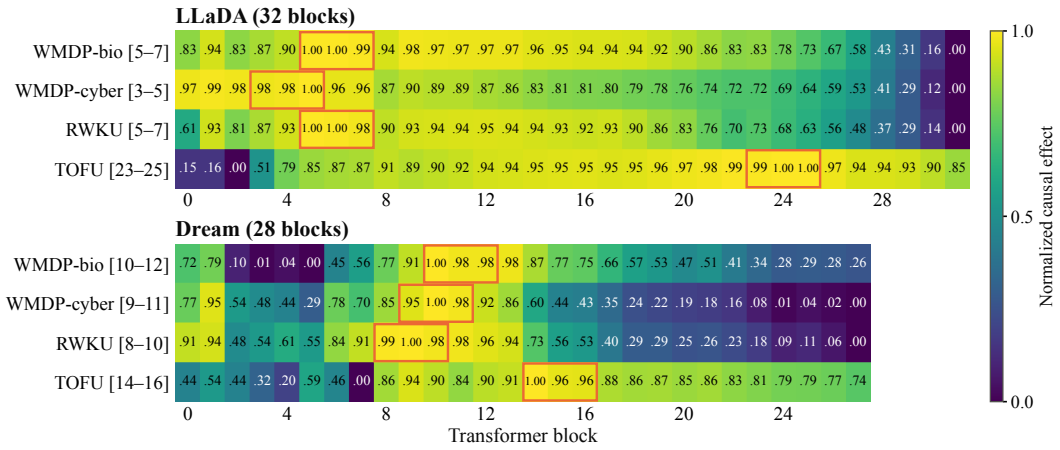


Figure 3: **Causal-mediation localization.** Colors and numbers show row-normalized answer recovery after restoring each block in a corrupted context. Orange windows and bracketed indices mark the selected three-block bands.

The probe changes activations without updating model weights or taking optimizer steps. After selection we discard the patched activations and train the chosen band from the original checkpoint. The recovery profiles in Figure 3 vary across backbones and knowledge domains, supporting a task-dependent choice of update location. This score measures activation recovery rather than performance after unlearning. We assess the trained model through target recall and retained capability, reporting both alongside their combined metric H (Section 4.1). Nearby-band ablations test how sensitive these outcomes are to the selected boundaries. Algorithm 1 summarizes the procedure, and Appendix C gives restoration details.

Computational cost. LRCU trains the attention, feed-forward and normalization parameters within the selected blocks. All parameters outside the band remain frozen. AdamW therefore stores optimizer state only for these trainable parameters. The per-step cost is approximately

$$C_{\text{step}} \approx C_{\text{forward}} + C_{\text{backward}}(a{:}L, W) + C_{\text{AdamW}}(P_W),$$

where a is the first trainable block, L is the final block, and P_W is the parameter count of band W. Backpropagation traverses blocks a through L but computes parameter gradients only within W. Thus depth determines the backward range while width determines how many parameters receive gradients and optimizer updates (Section 4.4). The method avoids a reference-model forward pass but still runs the full model forward, so updating fewer parameters does not reduce the cost proportionally.

4 Experiments

4.1 Experimental setup

Backbones and evaluation. We evaluate LLaDA-8B-Instruct and Dream-7B-Instruct with instruction-tuned checkpoints for WMDP and RWKU and separate supervised fine-tuning (SFT) checkpoints for TOFU. WMDP-bio and WMDP-cyber measure hazardous-knowledge recall on all 1,273 and 1,987 multiple-choice questions respectively (Li et al., 2024). We measure general ability with 30 questions from each of 57 MMLU subjects (1,710 total), reporting bio-adjacent subjects separately. RWKU evaluates ten independent entity edits across three forget and two neighbor splits, with at most 50 questions per entity and split (Jin et al., 2024). TOFU uses forget10 and retain90 with 100 questions per perturbed evaluation split (Maini et al., 2024). Both report ROUGE-L recall using native samplers with 64 denoising steps and at most 64 new tokens. With target recall F and retained performance U, both on $[0, 1]$, we define $H = 2(1 - F)U \,/\, (1 - F + U)$, with $H = 0$ when the denominator is zero. For WMDP these are domain and MMLU accuracy. RWKU averages the three forget splits and the two neighbor splits. TOFU uses forget10 and the mean of its three non-target splits. Higher H indicates a better forgetting–retention balance. Tables bold the best unlearned score and underline the second best, excluding base models.

Table 1: **WMDP forgetting and MMLU retention.** Bio / Other separate bio-adjacent and remaining MMLU subjects. Cyber uses all 1,987 questions.

		LLaDA					Dream				
Method	Params ↓	H ↑	Forget ↓	MMLU ↑	Bio ↑	Other ↑	H ↑	Forget ↓	MMLU ↑	Bio ↑	Other ↑
WMDP-bio											
Base	—	0.448	0.658	0.649	0.664	0.645	0.416	0.705	0.708	0.742	0.700
GA	100%	0.378	0.245	0.252	0.247	0.253	0.358	0.247	0.235	0.239	0.234
GD	100%	0.507	**0.241**	0.381	0.267	0.409	0.393	**0.240**	0.265	0.239	0.271
NPO	100%	0.547	0.417	0.516	0.522	0.515	0.471	0.341	0.367	0.369	0.367
SimNPO	100%	0.533	0.386	0.471	0.465	0.472	0.479	0.487	0.450	0.512	0.435
WGA	100%	0.534	0.498	0.571	0.593	0.565	0.504	0.545	0.564	0.558	0.566
MDU-style	100%	0.480	0.616	**0.640**	**0.685**	**0.630**	0.572	0.515	**0.696**	**0.730**	**0.688**
LRCU	**<10%**	**0.641**	0.293	0.587	0.476	0.613	**0.649**	0.312	0.614	0.585	0.621
WMDP-cyber											
Base	—	0.599	0.444	0.649	0.664	0.645	0.638	0.420	0.708	0.742	0.700
GA	100%	0.353	0.290	0.235	0.239	0.234	0.387	0.280	0.265	0.239	0.271
GD	100%	0.555	0.290	0.455	0.497	0.445	0.423	0.280	0.299	0.291	0.301
NPO	100%	0.617	0.393	0.627	0.660	0.620	0.509	0.330	0.410	0.406	0.411
SimNPO	100%	0.595	0.310	0.523	0.570	0.512	0.526	0.270	0.411	0.418	0.409
WGA	100%	0.621	0.310	0.565	0.605	0.555	0.588	0.380	0.559	0.552	0.561
MDU-style	100%	0.660	0.310	**0.632**	0.664	**0.624**	0.673	0.300	0.648	0.724	0.630
LRCU	**<10%**	**0.677**	**0.257**	0.621	**0.682**	0.607	**0.697**	**0.254**	**0.654**	**0.733**	**0.636**

Methods and training. We implement GA, GD, NPO, SimNPO, WGA, and an MDU-style anchor in the same framework (Zhang et al., 2024; Fan et al., 2025; Wang et al., 2025; Lee et al., 2026). Our anchor masks the entire input, whereas MDU masks only the prompt and retains the partially masked response. Retain terms use WikiText outside TOFU. Full-model NPO, SimNPO, and WGA use only forget data. Runs use 500 updates, two 512-token sequences per loss and 150 warm-up steps. LRCU trains three blocks with $c = \alpha = 1$ at learning rate 5×10^{-5}, versus 2×10^{-5} for full-model baselines. Fixed-band comparisons share LRCU's schedule and weighting at blocks 5–7 on LLaDA and 10–12 on Dream, with CE retention except for GA. Fixed-band and ablation results use separate runs. Benchmark runs use the bands in Figure 3. Appendix A covers optimization and alignment.

4.2 Main results

We first compare forgetting and retained capability across methods. The full-model baselines use 8-bit AdamW. Section 4.3 isolates the forget objective under shared training settings. In Tables 1 and 2, Params denotes the percentage of model parameters updated. LRCU updates 0.65B parameters (8.2%) on LLaDA and 0.70B (9.2%) on Dream.

Hazardous knowledge. On WMDP-bio, LRCU strikes the best balance between forgetting and utility, with H of 0.641 on LLaDA and 0.649 on Dream (Table 1). The baselines fail in opposite directions. GA and GD push forget accuracy to chance but pull MMLU to 0.235, whereas the MDU-style anchor preserves MMLU but leaves much target knowledge in place. LRCU lands between these extremes and gives up some target recall for most of the model's general capability. The full cyber split repeats this pattern, with near-chance forget accuracy and H of 0.677 and 0.697.

Biography recall. On TOFU, LRCU cuts forget recall by 74% on LLaDA and 91% on Dream while retain recall drops by at most 0.06 (Table 2, Appendix A). Its H is below the MDU-style anchor on LLaDA and matches it on Dream, with only three blocks updated. On LLaDA, the anchor's advantage comes mainly from lower forget recall with similar retain90 recall. Dream forgets more deeply at a larger cost to retention, so the balance point differs by backbone. GD suppresses target biographies more strongly than LRCU but loses more retained-biography recall on both models. These differences show why lower target recall alone does not establish a better unlearning outcome.

Table 2: **TOFU forget10 ROUGE-L recall.** Authors and Facts denote real-author and world-fact recall. GD, MDU-style and LRCU use retain90.

		LLaDA					Dream				
Method	Params ↓	H ↑	Forget ↓	Retain ↑	Authors ↑	Facts ↑	H ↑	Forget ↓	Retain ↑	Authors ↑	Facts ↑
Base SFT	—	0.245	0.851	0.898	0.430	0.757	0.506	0.614	0.626	0.698	0.883
GA	100%	0.000	**0.000**	0.000	0.000	0.000	0.000	**0.000**	0.000	0.000	0.000
GD	100%	0.703	0.010	0.786	0.341	0.509	0.425	0.016	0.497	0.290	0.025
NPO	100%	0.011	**0.000**	0.000	0.000	0.017	0.000	**0.000**	0.000	0.000	0.000
SimNPO	100%	0.000	**0.000**	0.000	0.000	0.000	0.000	**0.000**	0.000	0.000	0.000
WGA	100%	0.318	0.039	0.083	0.035	0.453	0.000	**0.000**	0.000	0.000	0.000
MDU-style	100%	**0.813**	0.033	**0.887**	**0.445**	0.770	0.802	0.051	0.558	0.662	**0.866**
LRCU	**<10%**	0.737	0.221	0.874	0.426	**0.798**	**0.804**	0.054	**0.567**	**0.686**	0.846

Table 3: **Forget objectives within a fixed band.** Methods share the band, weighting and schedule. All except GA use CE retention (Section 4.1).

	LLaDA (5–7)					Dream (10–12)				
Forget loss	H ↑	WMDP ↓	MMLU ↑	Bio ↑	Other ↑	H ↑	WMDP ↓	MMLU ↑	Bio ↑	Other ↑
GA	0.393	0.273	0.270	0.255	0.273	0.362	0.247	0.238	0.242	0.237
GD	0.394	0.250	0.267	0.242	0.273	0.362	**0.246**	0.238	0.242	0.237
NPO	0.563	**0.247**	0.449	0.361	0.471	0.442	0.256	0.314	0.282	0.322
SimNPO	0.602	0.251	0.503	0.375	0.533	0.451	0.266	0.326	0.336	0.323
WGA	0.661	0.300	**0.626**	**0.604**	**0.631**	0.586	0.306	0.507	0.496	0.509
MDU-style	0.617	0.264	0.530	0.418	0.557	0.482	0.259	0.357	0.318	0.367
Cap (LRCU)	**0.668**	0.273	0.618	0.572	0.629	**0.634**	0.288	**0.571**	**0.545**	**0.578**

Non-target knowledge. TOFU's real-author and world-fact splits test whether retention extends beyond the fine-tuning corpus. LRCU retains substantial non-target recall on both models. Forget-only GA, NPO, and SimNPO lose nearly all recall on both splits, GD loses world-fact recall despite its retain term, and WGA keeps some world-fact recall on LLaDA but collapses on Dream. For LRCU on LLaDA, real-author recall stays close to the starting score and world-fact recall remains above it. On Dream, both decline but stay above GD and NPO. Retention thus extends beyond retain90 but varies by knowledge type.

4.3 Why localize and why cap

Objective choice within a fixed band. To isolate objective choice, Table 3 fixes the band, weighting and schedule, with shared CE retention except for forget-only GA. NPO forgets more deeply than the cap but loses more utility on both backbones, so localization alone does not protect retained knowledge. The cap stops each token's forget gradient at the recall threshold, while GD and NPO continue suppressing it. WGA gives a similar trade-off on LLaDA through a smoothly decreasing weight. The cap achieves the highest H on both backbones, supporting explicit gradient stopping within the selected band.

Task-dependent update location. Figure 3 shows that the same causal rule selects different bands across models and domains. For WMDP-cyber, these are blocks 3–5 on LLaDA and 9–11 on Dream, compared with 23–25 and 14–16 for TOFU. Each band contains three blocks, so only update depth changes. WMDP-cyber bands sit earlier than TOFU bands on both backbones, so the rule adapts to the domain as well as the model. The selected depth follows where the probe finds recovery rather than a fixed layer range, which is why no single band serves every task. Nearby-band ablations also find useful forgetting in adjacent LLaDA windows (Table 7). On WMDP-bio, the causal band ranks ninth of 30 after full training (Appendix C). These results support task-dependent selection with some flexibility in band boundaries.

Table 4: **Component ablations.** WMDP-bio, 500 updates, seed 42.

Variant	$H \uparrow$	Forget ↓	MMLU ↑
LLaDA			
No cap	.391	.258	.265
16 blocks	.641	.285	.581
Uniform	.671	.268	**.619**
LRCU	**.675**	**.257**	**.619**
Dream			
No cap	.361	**.246**	.237
16 blocks	.615	.288	.541
Uniform	.614	.439	**.677**
LRCU	**.624**	.296	.561

Table 5: **LLaDA ms per step on one A100.** LRCU rows share training settings.

Method	Blocks	Forward	Backward	Optimizer	Total ↓
GA	All	**97**	186	233	522
GD	All	197	402	239	850
NPO	All	196	187	233	624
SimNPO	All	**97**	187	233	524
WGA	All	**97**	187	232	524
MDU-style	All	290	402	233	941
LRCU	All	194	396	233	835
LRCU	0–2	195	251	22	479
LRCU	5–7	195	215	**21**	440
LRCU	29–31	194	**48**	**21**	**268**

4.4 Ablation and analysis

Component ablation. Table 4 removes the cap, widens the band to blocks 0–15, or replaces timestep weighting with uniform weights, one change at a time. The cap matters most. Removing it sharply reduces utility on both backbones, whereas widening the band costs far less. Timestep weighting plays a different role. It changes how much each masking level contributes to training, and its effect shows mainly on forgetting in Dream. The separate width study points the same way, since several band sizes forget well (Appendix E). Gradient stopping is therefore the load-bearing component in this ablation, while width and weighting only adjust the resulting trade-off.

Training cost. Localization also makes training cheaper. On one A100, LRCU at blocks 5–7 takes 440 ms per step, whereas full-model baselines take 522–941 ms (Table 5). The cleanest comparison holds the cap and optimizer fixed and changes only the update region, which cuts the step from 835 to 440 ms. Most of the saving comes from the optimizer, whose time falls from 233 to 21 ms, and the rest from a shorter backward pass, which drops from 396 to 215 ms while the forward pass stays near 194 ms. Backward cost also depends on where the band sits, falling from 251 ms at blocks 0–2 to 48 ms at blocks 29–31. Reference-based methods pay further, since NPO and the MDU-style anchor add forward passes of 99 and 96 ms. LRCU therefore saves on parameter updates, on backward length, and on the extra forward pass.

5 Conclusion

Unlearning in DLMs must suppress target knowledge without eroding general capability. LRCU addresses this trade-off by combining causal-mediation localization with a per-token recall cap. Across two backbones, it improves the forgetting–utility harmonic mean H over the compared full-model recipes on WMDP and remains competitive on TOFU with a smaller update budget. It updates only 8.2–9.2% of parameters without a reference model, with training steps 1.2–2.1× faster than the full-model baselines on LLaDA. These results make restricted weight updates a practical option for DLM unlearning and show why localization and gradient stopping matter together.

Limitations and future work. The main limitation is the durability of forgetting. Five relearning updates restore LLaDA's WMDP-bio accuracy from 0.296 to 0.639 (Appendix H), showing that suppressed recall can recover quickly. Future work should test whether training against relearning can make this suppression more persistent. Band selection also relies on activation recovery, leaving room to improve how localization predicts the outcome of weight updates.

The recall bound concerns verbatim recovery under a probability ceiling across all reachable sampling histories. Training encourages this ceiling only in sampled contexts, and our generation evaluation uses the native samplers. Extending the analysis to other decoders and semantic recall would clarify how broadly the observed forgetting transfers.

AI USE STATEMENT

In this work, we used generative AI tools to assist with method implementation, experiment design and execution, interpretation of results, formulation of mathematical claims and proof writing. We did not use them to generate synthetic data or to clean datasets. Translation and qualitative analysis are not applicable. We also used them for software development, drafting and polishing the manuscript, figure preparation and literature search. We have reviewed all AI-assisted work, verified the code by testing and take responsibility for the final content of this work including text, claims and artifacts produced with generative AI.

ETHICS STATEMENT

We study knowledge suppression on public benchmarks of hazardous knowledge, real-world entities, and synthetic biographies. Lower recall does not establish model safety or irrecoverable removal of private information. The released checkpoints suppress recall under the evaluated conditions but should not be treated as models from which the knowledge has been durably removed. Our relearning results reinforce this limitation and motivate combining unlearning with access controls and deployment safeguards.

REPRODUCIBILITY STATEMENT

Section 3 gives the method and recall-bound assumptions. Section 4.1 gives the evaluation protocol. Appendices C, A, and B specify the probe budgets and implementation. We will release the code upon acceptance.

REFERENCES

Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023. URL https://arxiv.org/abs/2303.08112.

Tiwei Bie, Maosong Cao, Kun Chen, Lun Du, Mingliang Gong, Zhuochen Gong, Yanmei Gu, Jiaqi Hu, Zenan Huang, Zhenzhong Lan, Chengxi Li, Chongxuan Li, Jianguo Li, Zehuan Li, Huabin Liu, Lin Liu, Guoshan Lu, Xiaocheng Lu, Yuxin Ma, Jianfeng Tan, Lanning Wei, Ji-Rong Wen, Yipeng Xing, Xiaolu Zhang, Junbo Zhao, Da Zheng, Jun Zhou, Junlin Zhou, Zhanchao Zhou, Liwang Zhu, and Yihong Zhuang. LLaDA2.0: Scaling up diffusion language models to 100B. *arXiv preprint arXiv:2512.15745*, 2025. URL https://arxiv.org/abs/2512.15745.

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for LLM unlearning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 1540–1567, 2025. doi: 10.52202/085713-0058. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/02443e4e008231e0af0855f3cc70ed17-Abstract-Conference.html.

Ce Fang, Zhikun Zhang, Min Chen, Qing Liu, Lu Zhou, Zhe Liu, and Yunjun Gao. KUDA: Knowledge unlearning by deviating representation for large language models. In *Network and Distributed System Security Symposium*, 2027. URL https://arxiv.org/abs/2602.19275. To appear.

Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via Min-SNR weighting strategy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7441–7451, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Hang_Efficient_Diffusion_Training_via_Min-SNR_Weighting_Strategy_ICCV_2023_paper.html.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14389–14408, 2023. doi: 10.18653/v1/2023.acl-long.805. URL https://aclanthology.org/2023.acl-long.805/.

Metod Jazbec, Theo X. Olausson, Louis Béthune, Pierre Ablin, Michael Kirchhof, João Monteiro, Victor Turrisi, Jason Ramapuram, and Marco Cuturi. Learning unmasking policies for diffusion language models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2512.09106.

Zhuoran Jin, Pengfei Cao, Chenhao Wang, Zhitao He, Hongbang Yuan, Jiachun Li, Yubo Chen, Kang Liu, and Jun Zhao. RWKU: Benchmarking real-world knowledge unlearning for large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 98213–98263, 2024. doi: 10.52202/079017-3117. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/b1f78dfc9ca0156498241012aec4efa0-Abstract-Datasets_and_Benchmarks_Track.html.

Shervin Khalafi, Alejandro Ribeiro, and Dongsheng Ding. Unlearning in diffusion models: A unified framework with KL divergence and likelihood constraints. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=98xDF2hkIy.

Georu Lee, Seungwon Jeong, Hoki Kim, Jinseong Park, and Woojin Lee. Machine unlearning for masked diffusion language models. *arXiv preprint arXiv:2605.18253*, 2026. URL https://arxiv.org/abs/2605.18253.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew Bo Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, Bhrugu Bharathi, Ariel Herbert-Voss, Cort B Breuer, Andy Zou, Mantas Mazeika, Zifan Wang, Palash Oswal, Weiran Lin, Adam Alfred Hunt, Justin Tienken-Harder, Kevin Y. Shih, Kemper Talley, John Guan, Ian Steneker, David Campbell, Brad Jokubaitis, Steven Basart, Stephen Fitz, Ponnurangam Kumaraguru, Kallol Krishna Karmakar, Uday Tupakula, Vijay Varadharajan, Yan Shoshitaishvili, Jimmy Ba, Kevin M. Esvelt, Alexandr Wang, and Dan Hendrycks. The WMDP benchmark: Measuring and reducing malicious use with unlearning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 28525–28550, 2024. URL https://proceedings.mlr.press/v235/li24bc.html.

Pengxiang Li, Yefan Zhou, Dilxat Muhtar, Lu Yin, Shilin Yan, Li Shen, Soroush Vosoughi, and Shiwei Liu. Diffusion language models know the answer before decoding. In *The Fourteenth International Conference on Learning Representations*, pp. 135235–135253, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/daadbff4d4ea884ca3d9d389a1dfc61c-Paper-Conference.pdf.

Tianyi Li, Mingda Chen, Bowei Guo, and Zhiqiang Shen. A survey on diffusion language models. *arXiv preprint arXiv:2508.10875*, 2025. URL https://arxiv.org/abs/2508.10875.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. TOFU: A task of fictitious unlearning for LLMs. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=B41hNBoWLo.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17359–17372, 2022. doi: 10.52202/068431-1262. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

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*, volume 38, pp. 50608–50646, 2025. doi: 10.52202/085713-1689. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/48b383b24230e0e6e649d9c98dae4d8c-Abstract-Conference.html.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective

masked diffusion language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 130136–130184, 2024. doi: 10.52202/079017-4135. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/eb0b13cc515724ab8015bc978fdde0ad-Abstract-Conference.html.

Adi Shnaidman, Erin Feiglin, Osher Yaari, Efrat Mentel, Amit Levi, and Raz Lapid. Activation steering for masked diffusion language models. In *ReALM-GEN Workshop at the International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=JhKgHfpIF8.

Qizhou Wang, Jin Peng Zhou, Zhanke Zhou, Saebyeol Shin, Bo Han, and Kilian Q. Weinberger. Rethinking LLM unlearning objectives: A gradient perspective and go beyond. In *The Thirteenth International Conference on Learning Representations*, pp. 61897–61931, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/9ad307ab459fb93163248a0a56959307-Abstract-Conference.html.

Zhengtao Yao, Liuyang Song, Hongbo Zhang, Chenhao Wei, Haoyan Xu, Guang Yang, and Siheng Wang. TimeROME-DLM: Temporal causal tracing and low-rank inference-time knowledge editing for masked diffusion language models. *arXiv preprint arXiv:2606.12841*, 2026. URL https://arxiv.org/abs/2606.12841.

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. URL https://arxiv.org/abs/2508.15487.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=MXLBXjQkmb.

Localized Recall-Capped Unlearning for Diffusion Language Models

Supplementary Material

TABLE OF CONTENTS

A Implementation and backbone details 14

B Open-DLU and reproducibility 14

C Causal-mediation localization 15

D Recall constraints and generation 15

E Additional experimental checks 16

F Predictions across denoising steps and layers 17

G Autoregressive control 17

H Relearning 18

A Implementation and backbone details

Training. LRCU uses AdamW with learning rate 5×10^{-5}, 150 warm-up updates, and 500 updates in total. The learning rate stays constant after warm-up and we clip gradients to norm 1 throughout. Each update draws a forget batch and a retain batch of two 512-token sequences each. Each batch draws its own mask rate from $\mathcal{U}(0.1, 0.9)$ and applies it with token-level Bernoulli masking. On both main backbones we set $c = 1$, retain weight $\alpha = 1$, and Min-SNR parameter $\gamma = 2$. Full-model baselines instead use learning rate 2×10^{-5}, cosine decay after warm-up, and 8-bit AdamW. In contrast, the fixed-band loss comparison shares LRCU's optimizer, masking, weighting, and retain loss. In both protocols NPO and SimNPO use $\beta = 0.2$, and SimNPO uses zero offset. WGA uses exponent 1 except in the LLaDA fixed-band comparison where it uses exponent 2. The full-model GA, GD, NPO, SimNPO, and WGA recipes all use $1/t$ weighting. Our MDU-style anchor uses temperature 1, uniform weighting, and CE retention.

Scoring and generation. Both the training losses and the likelihood-based evaluation use the backbone-specific logit alignment defined in Section 3.1. Generation-based evaluation instead uses each backbone's native sampler with 64 denoising steps and at most 64 new tokens for RWKU and TOFU.

TOFU starting models. TOFU unlearning starts from a model fine-tuned on the full synthetic biography corpus. For LLaDA we use the SFT checkpoint released with Open-DLU. For Dream we use the 50-epoch checkpoint from our own SFT routine. This routine masks answer tokens but leaves the question visible. It computes the inverse-time-weighted loss only at the masked answer positions. It also excludes padding from valid-token attention and uses the same shifted target alignment as unlearning. Within each backbone, LRCU, GD, and NPO all start from the same SFT checkpoint. Table 2 reports the starting scores and unlearning results for each backbone.

RWKU evaluation. We follow the per-entity setup of Lee et al. (2026). Causal localization selects one band per backbone before the independent entity edits. For each of the first ten entities we then reuse that band, reset the model, unlearn that entity, and evaluate its forget and neighboring-entity questions. Each split uses at most 50 questions per entity. LRCU trains on each entity's introduction alone, so no evaluation question appears in its forget corpus. The final score averages the ten entity-level ROUGE-L recalls. Table 6 lists published baseline results next to our LRCU runs.

RWKU results and limitations. RWKU tests whether this balance carries over from choosing an answer to generating one (Table 6). The comparison includes forget-only baselines and LRCU with WikiText retention, so it captures the combined effect of the forget objective and retain loss. On LLaDA, LRCU reaches a mean target recall of 0.195 and a mean neighbor-entity recall of 0.373, for $H = 0.510$. It keeps more neighbor knowledge than the published GA and NPO results but removes less target recall. MDU reaches a higher H at $\tau = 1$ and almost complete forgetting at $\tau = 0$. On Dream, LRCU reaches a mean target recall of 0.396 and a neighbor recall of 0.406, for $H = 0.485$. It retains the most neighbor knowledge among the compared methods but leaves substantial target recall, with F-L1 exceeding the reported base score. RWKU exposes weaker target suppression, particularly on Dream, motivating further study of entity-level forgetting while preserving neighboring knowledge.

B Open-DLU and reproducibility

Open-DLU separates the forget loss, retain loss, timestep weighting, mask sampler, and parameter adapter to support controlled comparisons within one training loop. It also includes benchmark evaluators and the SFT routine for TOFU starting models. Each backbone–dataset pair uses a fixed training payload containing chunks, evaluation questions, reference answers, and a content hash. Run records store the seed, model source, configuration, per-step losses, per-question predictions, and trained band weights. We compute H before rounding its aggregate components. For each recipe we use the latest completed run per seed and do not pool across bands, protocols, or starting checkpoints. We release the model files and configurations to support reproduction.

Table 6: **RWKU target and neighbor recall.** H is the harmonic mean of one minus mean F-L and mean N-L. Entity-macro ROUGE-L over ten edits. Baselines use forget data alone while LRCU adds WikiText retention.

Method	$H \uparrow$	F-L1 $\downarrow$	F-L2 $\downarrow$	F-L3 $\downarrow$	N-L1 $\uparrow$	N-L2 $\uparrow$
LLaDA (LRCU: blocks 5–7)						
Base	0.496	0.488	0.433	0.469	0.551	0.372
GA	0.357	0.098	0.087	0.080	0.246	0.198
NPO	0.357	0.085	<u>0.074</u>	0.081	0.252	0.191
SimNPO	0.359	0.086	0.082	0.086	0.255	0.192
WGA	0.370	0.087	0.095	0.087	0.255	0.210
DPO	0.382	<u>0.058</u>	0.091	0.076	0.261	0.221
MDU ($\tau = 0$)	0.433	**0.015**	**0.006**	**0.015**	0.330	0.225
MDU ($\tau = 1$)	**0.539**	0.108	0.087	<u>0.072</u>	<u>0.431</u>	**0.334**
LRCU	<u>0.510</u>	0.236	0.137	0.210	**0.440**	<u>0.307</u>
Dream (LRCU: blocks 8–10)						
Base	0.487	0.418	0.438	0.457	0.504	0.356
GA	0.460	0.290	<u>0.188</u>	0.368	0.420	0.257
NPO	0.467	<u>0.193</u>	<u>0.188</u>	0.368	0.420	0.257
SimNPO	0.428	0.276	0.227	0.363	0.369	0.244
WGA	0.416	0.217	0.209	<u>0.253</u>	0.367	0.202
DPO	0.480	0.308	0.386	0.433	0.471	0.308
MDU ($\tau = 0$)	0.464	**0.168**	**0.152**	**0.218**	0.395	0.252
MDU ($\tau = 1$)	**0.514**	0.262	0.245	0.305	<u>0.475</u>	<u>0.318</u>
LRCU	<u>0.485</u>	0.435	0.333	0.420	**0.486**	**0.326**

C CAUSAL-MEDIATION LOCALIZATION

The localization probe follows the causal-mediation procedure in Section 3.3. Each probe uses a text chunk with its first 60% as context and the rest as masked targets. A clean pass gives the hidden states to restore and a corrupted pass gives the reference target log-probability. This corrupted pass also masks the context. We then restore only context-position states at one block and measure recovery at the masked target positions. The profiles use eight chunks per model and domain, except TOFU, which uses 32.

Figure 3 normalizes each model–domain row for display. The colors therefore show where recovery concentrates within a row rather than comparable effect sizes across domains. The orange windows mark the selected bands, with zero-based indices beside each dataset. To compare the selected band with other locations, we train every contiguous three-block LLaDA band on WMDP-bio for 500 updates with seed 42 (Figure 4). The causal band 5–7 ranks ninth of 30 by H and lies 0.016 below the best band, 7–9. The benchmark run of the same band reaches $H = 0.641$, so run-to-run variation is similar in size to this gap. The eleven highest-scoring bands all start within the first 12 blocks. In contrast, every band that starts at block 12 or later leaves forget accuracy at 0.49 or higher. Within this single-model, single-domain check, the causal band lies in a high-scoring region but is not optimal.

D RECALL CONSTRAINTS AND GENERATION

From recall constraints to a penalty. Equation (1) expresses unlearning as keeping useful predictions while enforcing a cross-entropy floor on forget tokens in given masked contexts. To obtain a training objective we penalize each constraint violation with the hinge

$$(c - \ell)_+ = \max\{c - \ell, 0\} = c - \min\{\ell, c\}. \tag{8}$$

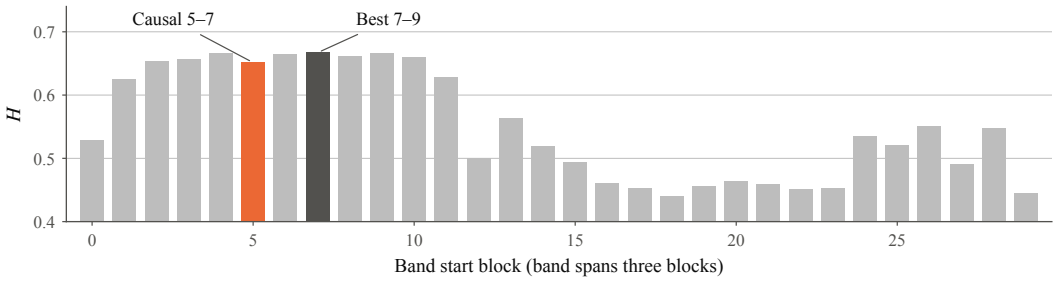


Figure 4: **All three-block bands on LLaDA WMDP-bio.** H after 500 updates with seed 42. The causal band is orange and the highest H is dark gray.

Up to a constant, this is the capped forget loss, which penalizes only the cross-entropy shortfall below c. This penalty is a tractable relaxation of the constraints. Because it is only a relaxation, finite training with a retain term does not guarantee that every constraint is satisfied.

Proof of Proposition 1. Fix the position j named in the proposition. Because the sampler fills every target position once, its final output at j is the token drawn after history H_j. Conditioning on this history gives

$$\Pr_\theta(X_j = y_j) = \mathbb{E}_{H_j}[p_\theta(y_j \mid H_j)] \le e^{-c}.$$

The inclusion $\{X_S = y_S\} \subseteq \{X_j = y_j\}$ then yields

$$\Pr_\theta(X_S = y_S) \le \Pr_\theta(X_j = y_j) \le e^{-c}.$$

The argument allows adaptive position ordering and does not assume independent token draws. □

Sampling conditions. The proof requires the ceiling to hold for every reachable history under the decoder's actual sampling distribution. This requirement is stronger than a ceiling on raw denoiser probabilities. Such a ceiling alone does not cover greedy decoding, transformed logits, or position selection based on sampled token values. Training on sampled corruptions encourages this condition but does not certify it for all generation histories.

E Additional experimental checks

MMLU coverage. The benchmark protocol covers all 57 MMLU subjects. The fixed-band and ablation runs cover 55–57 subjects because some subject-loading requests failed. In the main tables Bio and Other average the available subjects from 11 bio-adjacent and 46 remaining MMLU subjects, respectively.

Ablation and band sensitivity. The band-sensitivity runs use $c = 1$, Min-SNR weighting, 500 steps, and seed 42. MMLU scores average the 48 subjects shared across these runs. Table 7 varies the LLaDA-bio band from two to five blocks and shifts the three-block window by one position. The results support useful forgetting across several nearby bands and widths.

Table 7: **LLaDA band sensitivity.** H is the harmonic mean of $1-$forget and MMLU. WMDP-bio and shared-subject MMLU with seed 42.

Blocks	$H \uparrow$	WMDP-bio $\downarrow$	MMLU $\uparrow$
5–7	**0.674**	**0.257**	<u>0.617</u>
5–6	0.654	0.290	0.606
4–6	<u>0.673</u>	<u>0.266</u>	**0.622**
6–8	0.648	0.285	0.592
5–8	0.663	0.270	0.608
5–9	0.647	0.297	0.599

Training cost. Table 5 separates forward, backward, and optimizer time on one H100. Timings average 17 updates after three warm-up steps and exclude band selection and evaluation. Three blocks hold 8.2% of LLaDA's parameters and 9.2% of Dream's when each total counts the embeddings and the output head. Forward timings include reference-model calls where needed. Full-model runs use 8-bit AdamW and band runs use standard AdamW. Neither uses gradient checkpointing. The NPO [5–7] timing row uses a sigmoid gate on the NPO loss and KL retention.

F PREDICTIONS ACROSS DENOISING STEPS AND LAYERS

We trace LLaDA's vocabulary predictions by applying its final normalization and output head to each block's hidden state during denoising. The scored targets are the non-special tokens in the base model's generated completion. We hold these targets fixed when we score the unlearned model. Each model still follows its own denoising trajectory, so the readout measures predictions in the contexts that the model actually sees. Figure 1 uses the TOFU benchmark checkpoint at blocks 23–25 with $c = 1$ and seed 42. We trace four forget and four retain questions selected by post-unlearning answer recall, then choose the names of Hsiao Yun-Hwa and Anara Yusifova as an illustrative pair. Both base answers have ROUGE-L 1, falling to 0 for the forget answer while remaining 1 for the retain answer. Among the four retain candidates, this item has the smallest trajectory change and highest final base-token probability. Panel (a) uses a centered three-step weighted mean $(1/4, 1/2, 1/4)$ with raw endpoints. Panel (b) shows raw final-step layer readouts. The pair is not a probe-set average. Figure 5 separately averages five RWKU forget probes using a fixed-band checkpoint at blocks 29–31 with $c = 1$ and seed 42.

The heatmaps separate the denoising step from transformer depth. Along the step axis, base-model completion tokens can already receive substantial probability in the first forward pass. Along the depth axis, their layer-wise readouts sharpen near the output. After unlearning, these same tokens receive less probability across the denoising trajectory. These readouts track prediction changes after unlearning, complementing the activation interventions used to select an update band.

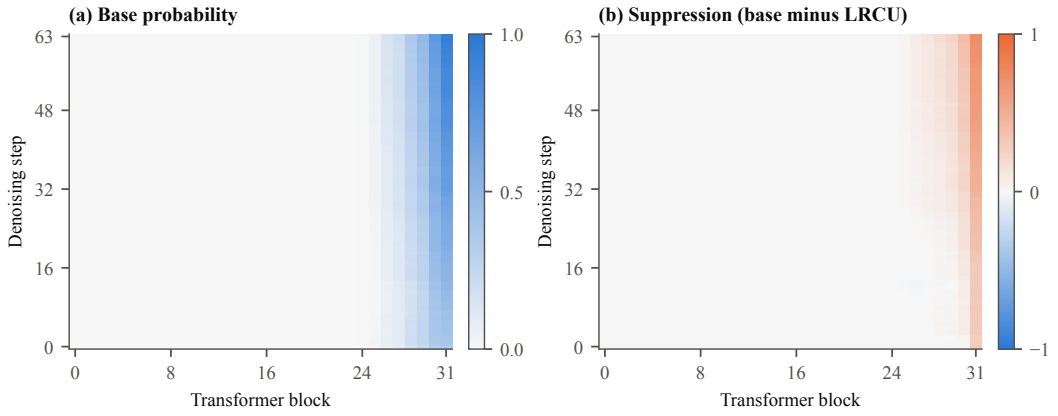


Figure 5: **Layer readouts over denoising.** Mean logit-lens probability of base-completion tokens over five RWKU forget probes. The left panel shows the base model. The right panel shows base minus LRCU, where positive values indicate suppression.

G AUTOREGRESSIVE CONTROL

The per-token cap does not require a diffusion model. We test this by applying it to teacher-forced token losses in Qwen2.5-7B-Instruct and updating blocks 10–17 with WikiText retention. All variants use 600 updates, learning rate 2×10^{-5}, retain weight 1, and seed 42. Evaluation uses 150 WMDP-bio questions and 12 MMLU questions per subject. Under this setup Table 8 shows the same qualitative benefit from stopping the forget gradient. With $c = 4$, target accuracy falls to 0.300 while MMLU remains at 0.685. Uncapped ascent reaches a lower target accuracy of 0.273 but leaves MMLU at

0.247. This control suggests that the stopping rule can preserve utility beyond DLMs. The generation bound for DLMs additionally requires the sampling conditions in Proposition 1.

Table 8: **Capped forgetting in Qwen2.5-7B-Instruct.** H is the harmonic mean of $1-$forget and MMLU. Blocks 10–17, CE retention, 600 updates, seed 42.

Method	Cap	$H \uparrow$	WMDP-bio $\downarrow$	MMLU $\uparrow$
Base	—	0.453	0.673	0.735
Cap	1	<u>0.554</u>	0.553	**0.727**
Cap	4	**0.692**	<u>0.300</u>	<u>0.685</u>
Uncapped ascent	∞	0.369	**0.273**	0.247

H RELEARNING

Full-model relearning. We fine-tune each unlearned LLaDA model on the WMDP-bio forget corpus for 40 full-model updates at learning rate 5×10^{-5}. Table 9 reports target accuracy after 5, 10, 20, and 40 updates with seed 42 throughout. An LRCU checkpoint at blocks 5–7, trained separately from the benchmark runs, recovers from 0.296 to 0.639 within five updates. This value approaches the original model's 0.658 accuracy. NPO recovers more slowly and reaches 0.625 after 40 updates. Only GD remains near chance, but its final non-bio MMLU is also near chance at 0.234. Low recovery alone therefore cannot distinguish durable forgetting from a loss of general capability.

Table 9: **LLaDA recall after relearning.** WMDP-bio accuracy by update count with non-bio MMLU at update 40 in the final column. Seed 42.

Method	0	5	10	20	40	Final MMLU $\uparrow$
NPO	0.363	0.247	0.247	0.266	0.625	0.617
GD	0.240	0.247	0.246	0.251	0.247	0.234
LRCU	0.296	0.639	0.644	0.643	0.636	0.634