CoLU: Confidence-Weighted Layerwise Unlearning for Large Language Models

Anonymous authors
Paper under double-blind review

Abstract

Large language model (LLM) unlearning aims to forget certain information while preserving others, without retraining from scratch. SoTA methods such as NPO, SimNPO, and SatImp balance forgetting and retain through whole model fine-tuning, while FOM-UL explores selective layer updates. Prior works show that suppressing target information can damage retained knowledge. This trade-off can vary across layers even under the same unlearning loss. In this work, we propose Confidence-Weighted Layerwise Unlearning (CoLU), which combines a novel confidence-weighted forgetting and retain regularization while updating a single selected transformer layer, rather than the whole model for unlearning. To reduce CoLU compute cost, we further propose Brief Layer Screening (BLS), which uses a newly defined normalized forgetting–retention(NFR) score to first rank layers through a brief unlearning trials using a subset of forgetting dataset and fewer epochs. BLS pre-selects top-k ranked layer candidates, which will then be fully profiled using full data analysis. BLS is proven to reduce the runtime of CoLU by 60.2-79.7% with almost the same performance. On TOFU dataset, CoLU improves both forgetting and retention over NPO, SatImp, and GA-based FOM-UL in all six TOFU settings, and over SimNPO in five of the six settings. On MUSE Books and News datasets, CoLU achieves the highest retained KnowMem and the PrivLeak closest to zero among the compared SOTA unlearning methods. CoLU presents a new record performance in both unlearning and retain in most settings up to date. Moreover, across the measured settings, average CoLU GPU memory usage is 77.4% to 79.5% lower than full-model unlearning.

1 Introduction

Large language models can memorize private or sensitive information and reproduce it in their responses (Jang et al., 2023; Carlini et al., 2021; Nasr et al., 2025). Once this information has been used for training, deleting that data from the training set does not remove its influence on the model (Bourtoule et al., 2021). Machine unlearning aims to reduce its influence on the model while preserving useful capabilities (Kurmanji et al., 2023; Liu et al., 2025). Retraining without the data that is to be forgotten is costly for large models and so methods that modify an existing model using forget and retain data have been proposed (Shi et al., 2025).

A common approach to unlearning is gradient ascent(GA) on the loss of the forget data, which lowers the likelihood of targeted responses (Yao et al., 2024). Although this can suppress unwanted outputs, continued updates can also damage performance on retained data (Zhang et al., 2024). NPO controls the forgetting process through a negative-preference objective, while SimNPO removes the reference model and accounts for response length (Zhang et al., 2024; Fan et al., 2025). These methods aim to improve forgetting while limiting degradation of retained performance, but effective unlearning still requires assessing whether reduced target likelihood is accompanied by preserved performance on other inputs. Even after model editing suppresses a target response, extraction attacks may recover the underlying information (Patil et al., 2024).

Token-level reweighting offers finer control over the forgetting signal. WGA (Wang et al., 2025b) downweights low-probability target tokens while SatImp (Yang et al., 2025) combines saturation- and importance-based token reweighting. The effect of these updates also depends on which parameters are allowed to change. Prior work has explored selective updates, including single-layer

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.

interventions and constrained optimization of selected LLM layers (Cai et al., 2025; Agrawal & Huang, 2025). This motivates us combining probability-weighted forgetting with a restricted update scope and an explicit retain objective. Choosing the layer is part of this design: we seek a location where the unlearning procedure improves forgetting while preserving retained behavior.

In this paper, we propose **Co**nfidence-Weighted **L**ayerwise **U**nlearning (CoLU), which combines probability-weighted forgetting and retain regularization while updating a single selected transformer layer. We use Confidence-Weighted Ascent (CWA), building on the power reweighting in WGA and SatImp (Wang et al., 2025b; Yang et al., 2025), to reduce each target token's weight as its probability falls. Our contributions are as follows:

- We propose CoLU, which combines confidence-weighted ascent and retain regularization while updating only one selected transformer layer.
- To reduce CoLU compute cost, we further propose Brief Layer Screening (BLS), which uses a newly defined normalized forgetting–retention(NFR) score to first rank layers through a brief unlearning trials using a subset of forgetting dataset and fewer epochs. BLS pre-selects top-k ranked layer candidates, which will then be fully profiled using full data analysis. BLS is proven to reduce the runtime of CoLU by 60.2-79.7% with almost the same performance.
- On TOFU (Maini et al., 2024), CoLU improves both forgetting and retention over NPO, SatImp, and FOM-UL (GA) in all six settings, and over SimNPO in five. On MUSE Books and News (Shi et al., 2025), CoLU achieves the highest retained KnowMem and the PrivLeak closest to zero among the SOTA unlearning methods. To summarize, in this work, CoLU reports a new record performance in both forgetting and retain up to date.
- Ablation studies show that layerwise updates improve NFR for GA, NPO, and SatImp on Phi-1.5 Forget-5%, and that CoLU improves both forgetting and retention over full-model SatImp with multiple candidate layers. Across the measured TOFU and MUSE settings, single-layer unlearning reduces average GPU memory usage by 77.4–79.5% relative to full-model unlearning.

2 Related Work

Machine unlearning. Machine unlearning aims to remove the influence of designated training data while preserving performance on data that should be retained. Retraining without the designated data provides a reference for this goal, but its computational cost motivates approximate methods that modify an existing model. In LLMs, these methods include direct fine-tuning, localized parameter modification, and approaches that use auxiliary models (Geng et al., 2025). A central concern is whether the intervention suppresses targeted information while preserving the model's other capabilities.

Among direct optimization methods, gradient ascent (GA) increases the prediction loss on forget data to reduce the likelihood of targeted responses. Continued ascent can also degrade retained performance, motivating objectives that control the strength of forgetting (Zhang et al., 2024). Negative Preference Optimization (NPO) uses a reference-relative sequence objective to slow the utility collapse associated with GA (Zhang et al., 2024). SimNPO removes the reference model and normalizes loss by response length to address reference-model bias and uneven allocation of unlearning updates (Fan et al., 2025). WGA (Wang et al., 2025b) and SatImp (Yang et al., 2025) use loss reweighting to adjust how much individual tokens contribute to unlearning on top of the basic GA. Both methods use power re-weighting but weight the token probabilities differently.

Selective parameter updates. Selective updates can improve unlearning efficiency and control. SLUG selects layers using gradient importance and forget–retain alignment, then tunes the step size along a gradient computed once at the original model (Cai et al., 2025). Selective Amnesia uses causal mediation analysis to guide layer-specific LLM optimization (Agrawal & Huang, 2025). SURE combines module-level gradient saliency with a large learning rate to address the recovery of forgotten information after quantization (Zhang et al., 2025). FOM-UL selects transformer layers using a forget-to-retain significance score (Ranjan et al., 2026).

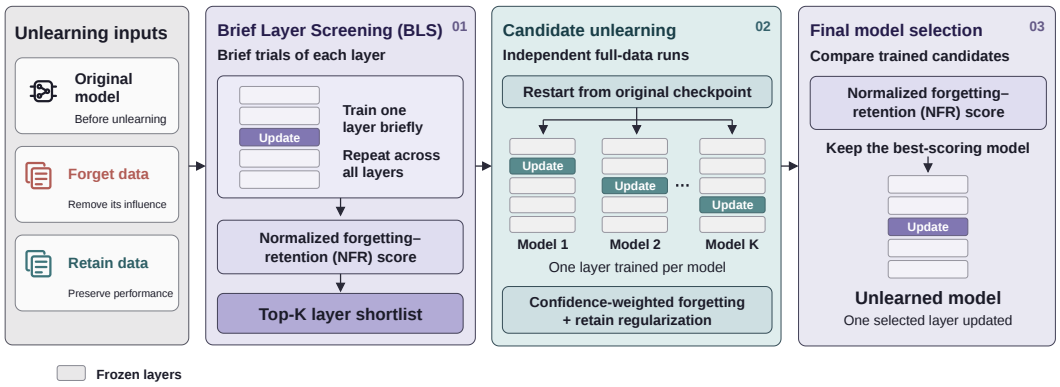


Figure 1: Overview of Confidence-Weighted Layerwise Unlearning (CoLU). Brief Layer Screening (BLS) ranks single-layer trials with less epochs using the normalized forgetting–retention (NFR) score and keeps the top-K layers. Each shortlisted candidate layer is trained independently while keeping the parameters of all other layers frozen. The same score selects the final model.

3 METHOD

We propose *Confidence-Weighted Layerwise Unlearning* (CoLU), it rank layers through a brief unlearning trials using a subset of forgetting dataset and fewer epochs. BLS pre-selects top-k ranked layer candidates, which will then be fully profiled using full data analysis. Finally, we select the model with the highest NFR score.

Confidence-Weighted Ascent loss. We use a *Confidence-Weighted Ascent* (CWA) loss, a scaled variant adoption of WGA (Wang et al., 2025b) that gives more weight to target tokens that the model still predicts with high probability. Let p_θ be the language model, with forget set $\mathcal{D}_f$ and retain set $\mathcal{D}_r$. For a target sequence y_i with context x_i, let $p_{i,t}$ denote the probability of token $y_{i,t}$, which we use as the token's confidence. Let $\mathcal{T}_i$ be the set of valid prediction positions for example i, and define $N_s = \sum_{i \in \mathcal{D}_s} |\mathcal{T}_i|$ for $s \in \{f, r\}$. These positions follow the causal next-token shift and exclude padding and masked context tokens.

Following the similar power weighting used in WGA and SatImp (Wang et al., 2025b; Yang et al., 2025), we define the CWA loss on the forget set $\mathcal{D}_f$ as

$$\mathcal{L}_f(\theta; \mathcal{D}_f) = \frac{1}{N_f} \sum_{i \in \mathcal{D}_f} \sum_{t \in \mathcal{T}_i} \mathrm{sg}\big(A p_{i,t}^{\alpha}\big) \log p_{i,t}, \qquad p_{i,t} = p_\theta(y_{i,t} \mid x_i, y_{i,<t}), \tag{1}$$

where $A > 0$ sets the scale and $\alpha > 0$ controls how strongly the weights favor higher-probability tokens. The stop-gradient operator sg holds the weights fixed during backpropagation, so gradients pass only through $\log p_{i,t}$. Minimizing the CWA loss performs weighted gradient ascent on the target-token negative log-likelihood. The loss focuses on confident predictions and reduces the impact of low-confidence ones. As a token's probability falls, its weight becomes smaller, reducing its contribution to later updates. A larger α makes this reduction sharper and also lowers the overall weight magnitude.

To preserve retain-set performance, we use retain regularization based on cross-entropy on $\mathcal{D}_r$ (Zhang et al., 2024; Fan et al., 2025), yielding the combined loss:

$$\mathcal{L}(\theta; \mathcal{D}_f, \mathcal{D}_r) = \lambda_f \mathcal{L}_f(\theta; \mathcal{D}_f) - \lambda_r \frac{1}{N_r} \sum_{i \in \mathcal{D}_r} \sum_{t \in \mathcal{T}_i} \log p_{i,t}. \tag{2}$$

The positive coefficients λ_f and λ_r set the strength of the two terms. We minimize the combined objective over the selected layer's parameters.

Layer Update scope. We use "layer" to represent a complete transformer block with self-attention, a feed-forward network, normalization, and residual connections. The original model has parameters θ_0 and L layers indexed from 0 to $L-1$. Each run updates all parameters in one candidate layer ℓ and freezes the rest of the model, including the token embeddings and output head.

Brief Layer Screening. To limit the cost of evaluating candidate layers, BLS measures how each responds to a short unlearning run. Every trial starts from θ_0 and uses the same sampled data and training budget within each experimental setting. On both TOFU and MUSE, we train on a fixed 20% subset of the training data for two epochs. We score the resulting models on fixed forget and retain screening subsets. And we use same evaluation data for all candidate layers, so the scores are comparable across layers.

We select candidate layers based on both forgetting and retention. Let $F_\ell^{(m)}$ denote the m-th forgetting metric after screening layer ℓ, for $m = 1, \ldots, M$, and let R_ℓ denote the retention metric. Lower forgetting scores and higher retention scores are preferred. We define the NFR score as

$$s_\ell = \frac{100}{2}\left[\frac{1}{M}\sum_{m=1}^{M}\left(1 - \frac{F_\ell^{(m)}}{F_O^{(m)}}\right) + \frac{R_\ell}{R_O}\right], \tag{3}$$

where $F_O^{(m)}, R_O > 0$ are the original model's scores for the same split. We first average the relative gains across the M forgetting metrics, then average this forgetting gain with normalized retention. Thus, forgetting and retention each receive half of the total weight, regardless of the number of forgetting metrics. For TOFU, $M = 1$ with forget ES and retain ES. For MUSE, $M = 2$ with forget-set VerbMem and KnowMem, and retention is measured by retain-set KnowMem. We leave the score unclipped, so it can fall outside the nominal 0–100 range. We use the same reference scores for every candidate and return the K highest-scoring layers, breaking ties by ascending layer index. We set $K = 3$ in the screening experiments.

Combined procedure. After BLS, we perform candidate unlearning that involves full-data unlearning independently for each of the K shortlisted candidate layers. Each run restarts from the original checkpoint and updates only the candidate layer using CWA and retain regularization, while all other parameters remain fixed. We then evaluate the resulting models using the same NFR score as in screening and report the results of the highest-scoring model.

Full-screening reference. To assess BLS selection, we use full screening (FS) as a reference for the best observed NFR across layers. FS performs full-data unlearning separately for each of the L layers. Each run starts from θ_0, updates one layer using the combined objective in Eq. 2, and freezes all other parameters. We select the resulting model with the highest NFR score.

4 Experimental Setup

Datasets and models. We evaluate our method on TOFU (Maini et al., 2024) and MUSE (Shi et al., 2025). TOFU contains 200 fictitious author profiles, each described by 20 question–answer pairs. We consider its three standard forget splits. Our TOFU experiments use the benchmark-provided Phi-1.5 and Llama-2-7B checkpoints, fine-tuned on the full TOFU dataset. For MUSE, we evaluate on both Books and News using the released target checkpoints, based on ICLM-7B and Llama-2-7B, respectively. Each checkpoint is fine-tuned on the corresponding domain's forget and retain data.

Evaluation metrics. For TOFU, our primary metric is extraction strength (ES), following the robustness analysis of Wang et al. (2025a). ES measures memorization through the shortest answer prefix needed to recover the remaining answer. Lower forget-set ES indicates less recoverable target information, while higher retain-set ES indicates better retention. We also report TOFU's Forget Quality (FQ) and Model Utility (MU) for completeness. FQ compares the forget-set truth-ratio distributions of the unlearned model and a retain-only reference using a Kolmogorov–Smirnov test. Following Yang et al. (2025), we distinguish agreement with this reference from reduced target-information extractability. A low FQ therefore does not, by itself, indicate failure under our forgetting objective. For both MUSE domains, we follow the benchmark's evaluation protocol and report verbatim memorization (VerbMem) and knowledge memorization (KnowMem) on the forget set, knowledge memorization on the retain set as a measure of utility, and privacy leakage (PrivLeak) (Shi et al., 2025). Lower forget-set memorization and higher retained utility are preferred. PrivLeak values closer to zero indicate closer agreement with the retain-only reference model under the benchmark's membership-inference evaluation.

Baselines. We compare CoLU with full-model baselines, including GA, NPO (Zhang et al., 2024), SimNPO (Fan et al., 2025), and SatImp (Yang et al., 2025), on TOFU and MUSE. On TOFU, we

Table 1: Comparison of different unlearning methods on TOFU. ↑ / ↓ indicate that larger / smaller values are preferable. Best ES and MU results are bold; second-best are underlined. FQ is not highlighted as it is less meaningful for unlearning. FU denotes FOM-UL(Ranjan et al., 2026).

Method	Forget-1%					Forget-5%					Forget-10%				
	Param.	ES Un.↓	ES Re.↑	FQ↑	MU↑	Param.	ES Un.↓	ES Re.↑	FQ↑	MU↑	Param.	ES Un.↓	ES Re.↑	FQ↑	MU↑
Phi-1.5															
Original	—	0.626	0.671	0.003	0.502	—	0.699	0.671	6.5e-13	0.502	—	0.704	0.671	2.2e-16	0.502
Retrain	All	0.041	0.756	1.000	0.498	All	0.053	0.756	1.000	0.505	All	0.054	0.782	1.000	0.496
GA	All	0.099	0.205	0.766	0.439	All	0.147	0.293	4.4e-8	0.455	All	0.099	0.166	4.3e-9	0.391
NPO	All	0.042	0.459	0.990	0.485	All	0.058	0.476	0.866	0.486	All	0.063	0.452	0.970	0.477
SimNPO	All	0.224	0.616	0.003	0.504	All	0.164	0.580	2.4e-8	0.494	All	0.189	0.559	1.4e-12	0.496
SatImp	All	0.054	0.571	0.097	**0.512**	All	0.038	0.521	0.040	0.496	All	0.035	0.458	0.007	0.509
CWA	All	**0.016**	0.518	0.579	0.506	All	0.010	0.496	0.142	0.507	All	0.037	0.501	1.2e-4	0.490
FU (GA)	Mul	0.155	0.593	0.007	0.503	Mul	0.249	0.448	6.6e-12	0.478	Mul	0.317	0.411	3.7e-15	**0.513**
FU (CWA)	Mul	0.024	0.574	0.650	0.499	Mul	0.039	0.532	0.051	0.495	Mul	0.012	0.485	2.2e-4	0.478
CoLU (FS)	L20	0.036	**0.636**	0.054	0.504	L4	0.020	**0.586**	0.088	**0.508**	L3	0.003	**0.579**	2.6e-9	0.501
CoLU (BLS)	L20	0.036	**0.636**	0.054	0.504	L23	**0.008**	0.550	0.012	0.476	L2	**0.002**	0.560	2.2e-6	0.493
Llama-2-7B															
Original	—	0.903	0.871	1.9e-4	0.587	—	0.876	0.871	3.4e-16	0.587	—	0.885	0.871	2.2e-20	0.587
Retrain	All	0.072	0.892	0.919	0.567	All	0.066	0.897	0.965	0.575	All	0.065	0.898	0.848	0.567
GA	All	0.057	0.303	0.405	0.468	All	**0.000**	0.160	3.0e-13	0.409	All	**0.000**	0.370	1.7e-26	0.529
NPO	All	0.055	0.662	0.990	0.580	All	0.047	0.673	0.713	0.590	All	0.049	0.596	0.293	0.555
SimNPO	All	0.132	0.746	0.014	0.598	All	0.133	0.687	0.001	0.595	All	0.171	0.626	1.2e-8	0.586
SatImp	All	0.066	0.787	0.165	0.567	All	0.006	0.709	3.1e-12	0.602	All	0.019	0.653	5.6e-14	0.588
CWA	All	0.019	0.705	0.097	**0.627**	All	0.005	0.677	4.7e-24	**0.637**	All	0.028	0.635	2.3e-7	**0.612**
FU (GA)	Mul	0.089	0.559	0.014	0.516	Mul	0.271	0.560	1.4e-11	0.435	Mul	0.408	0.585	1.8e-15	0.511
FU (CWA)	Mul	0.069	0.779	0.165	0.598	Mul	0.007	0.739	0.221	0.583	Mul	0.005	0.602	5.1e-17	0.604
CoLU (FS)	L4	**0.019**	**0.856**	0.405	0.575	L3	0.003	**0.833**	1.3e-13	0.602	L3	0.004	**0.849**	1.8e-15	0.600
CoLU (BLS)	L4	**0.019**	**0.856**	0.405	0.575	L3	0.003	**0.833**	1.3e-13	0.602	L3	0.004	**0.849**	1.8e-15	0.600

also compare with the multi-layer method FOM-UL (Ranjan et al., 2026). We evaluate two variants, denoted FU (GA) and FU (CWA) in Table 1. FU (GA) uses the original paper's GA-based objective, while FU (CWA) uses CWA loss for forgetting. We also include the Original and Retrain models as references for both datasets.

Throughout the tables, Param. indicates which parameters are updated: All denotes the full model, Lj denotes transformer layer j (using zero-based indexing), and Mul denotes progressive updates across multiple layers. A dash indicates no parameter updates.

Implementation and hyperparameters. CoLU full-data runs use five epochs on TOFU and eight epochs on MUSE, for both FS and the BLS shortlisted candidates unlearning. On both benchmarks, BLS screens each layer for two epochs on a fixed 20% subset of the training data and selects $K = 3$ candidates for full-data unlearning. BLS uses the same optimizer, batch size, and loss hyperparameters as the corresponding full-data runs, with a higher learning rate. We report dataset-specific hyperparameters in Appendix A.

5 Results and Analysis

Performance on TOFU. Table 1 compares TOFU results across both models and all three forget splits. CoLU (BLS) improves both forget and retain ES over SatImp in all six settings and over SimNPO in five. On Llama-2-7B Forget-10%, its retain ES reaches 0.849, dramatically exceeding SatImp and SimNPO by 19.6 and 22.3 percentage points, respectively, while its forget ES is 0.004, compared with 0.019 and 0.171. The exception is Phi-1.5 Forget-5%, where CoLU (BLS) reduces forget ES from SimNPO's 0.164 to 0.008 but has lower retain ES (0.550 versus 0.580).

The full-screening reference, CoLU (FS), achieves the highest retain ES among the compared unlearning methods in all six settings, together with lower forget ES than NPO, SimNPO, SatImp, and FU (GA). CoLU (BLS) selects the same layer and matches this reference in four settings, including

Table 2: Unlearning results on MUSE Books and News. VerbM/KnowM denote VerbMem/KnowMem; F/R denote forget/retain sets. ↑ / ↓ indicate that larger / smaller values are preferable; → 0 indicates closing to zero are preferable. Best values are bold; second-best are underlined.

Method	Books					News				
	Param.	VerbM F↓	KnowM F↓	KnowM R↑	PrivLeak → 0	Param.	VerbM F↓	KnowM F↓	KnowM R↑	PrivLeak → 0
Original	—	99.56	58.32	67.01	-56.32	—	58.29	62.93	54.31	-98.71
Retrain	All	14.30	28.90	74.50	0.00	All	20.75	33.32	53.79	0.00
GA	All	**0.00**	**0.00**	0.00	-24.07	All	**0.00**	**0.00**	0.00	<u>20.14</u>
NPO	All	**0.00**	**0.00**	23.71	-31.17	All	2.53	56.93	37.58	108.91
SimNPO	All	**0.00**	**0.00**	<u>48.27</u>	<u>-19.82</u>	All	2.34	44.84	<u>39.65</u>	72.93
SatImp	All	**0.00**	<u>1.38</u>	39.52	-62.19	All	<u>1.93</u>	<u>2.75</u>	17.62	-27.31
CoLU (FS)	L28	**0.00**	**0.00**	**61.85**	**-19.06**	L25	4.63	12.81	**43.01**	**-12.28**
CoLU (BLS)	L28	**0.00**	**0.00**	**61.85**	**-19.06**	L25	4.63	12.81	**43.01**	**-12.28**

all three Llama-2-7B splits. On Phi-1.5 Forget-5% and Forget-10%, it achieves lower forget ES with lower retain ES than FS. CoLU (BLS) therefore preserves much of the performance of full screening while limiting full-data unlearning to the three BLS-shortlisted candidates.

We include full-model CWA to examine the benefit of layerwise updates. Full-model CWA achieves lower forget ES than CoLU (BLS) only on Phi-1.5 Forget-1%, but with much lower retain ES. The consistent retain ES gains support combining CWA with layer selection to preserve retain-set behavior during unlearning while achieving similar or lower forget ES in most settings.

Performance on MUSE. Table 2 compares results on MUSE Books and News. BLS selects the same layer as FS in both domains. On Books, CoLU records 0.00 on both forget-set memorization metrics while achieving a retained KnowMem of 61.85. This exceeds SimNPO by 13.58 percentage points and our SatImp reproduction by 22.33 points. CoLU also achieves the PrivLeak closest to zero among the SOTA methods, at -19.06.

On News, CoLU achieves the highest retained KnowMem at 43.01, compared with 39.65 for SimNPO and 17.62 for SatImp. Its forget-set KnowMem is 12.81, well below SimNPO's 44.84, and its PrivLeak of -12.28 is again the closest to zero among the compared unlearning methods.

Together, the results on both MUSE domains show that CoLU reduces forget-set memorization while achieving the highest retained KnowMem and the PrivLeak closest to zero among the compared unlearning methods.

6 Ablation Studies

6.1 Layerwise Comparison Between Methods

Table 3: Layerwise and full-model unlearning on Phi-1.5 with the TOFU Forget-5% split. ↑ / ↓ indicate that larger / smaller values are preferable. Layerwise results are picked best from the full sweep of all layers using the NFR score in Eq. (3). The best value in each metric column is bold.

Method	Layerwise				Full model		
	Layer	Forget ES↓	Retain ES↑	NFR↑	Forget ES↓	Retain ES↑	NFR↑
GA	0	0.201712	0.513161	73.81	0.146975	0.293013	61.32
NPO	0	0.104086	0.593343	86.76	0.057829	0.475640	81.30
SatImp	2	0.092705	**0.598647**	87.97	0.038465	**0.521383**	86.10
CWA (ours)	4	**0.020209**	0.586068	**92.22**	**0.010440**	0.495960	**86.21**

To illustrate the performance of layerwise unlearning on different loss functions, Table 3 compares layerwise unlearning with full-model results on Phi-1.5 using the TOFU Forget-5% split. For the layerwise runs, we screen all layers and select the best-performing one for each method using the NFR score to show its upper bound performance. The layerwise configurations raise NFR from

61.32 to 73.81 for GA, from 81.30 to 86.76 for NPO, and from 86.10 to 87.97 for SatImp. All three achieve better retention despite higher forget ES, with GA showing the largest NFR gain. These results show that selected-layer versions of existing objectives can improve the forgetting–retention trade-off, although all three remain below CoLU in NFR score. Among layerwise results, CoLU achieves the lowest forget ES and the highest NFR score, while SatImp achieves the highest retain ES. Full-model CWA achieves lower forget ES than layerwise CWA, while the layerwise run achieves higher retain ES and raises the NFR score from 86.21 to 92.22.

6.2 Full-Screening Results

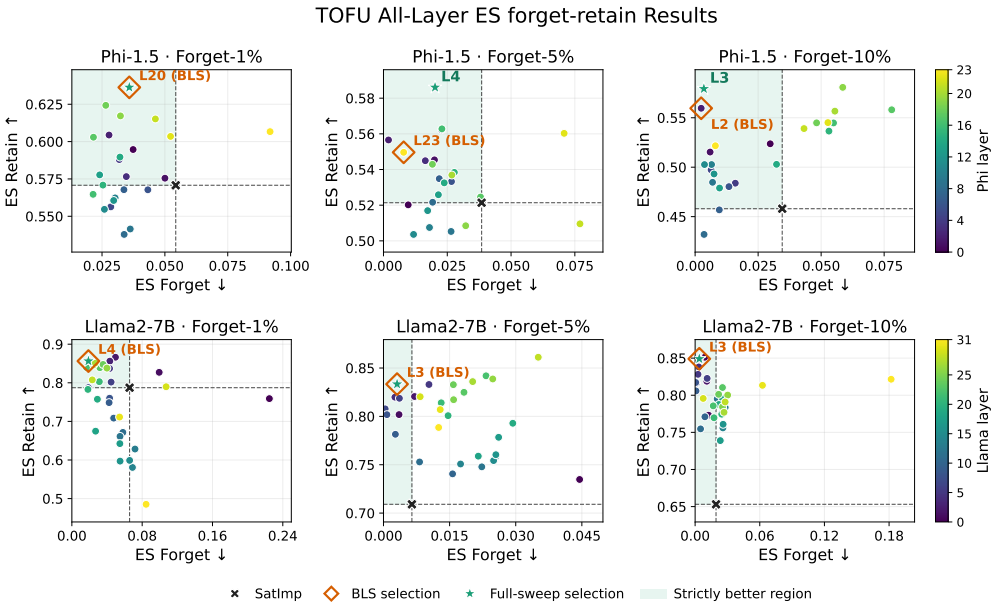


Figure 2: Full-screening results for CoLU on TOFU. Rows show Phi-1.5 and Llama-2-7B; columns show Forget-1%, Forget-5%, and Forget-10%. Each point represents an independently trained layer, colored by layer index.

Figure 2 plots forget ES and retain ES after applying CoLU independently to all 24 Phi-1.5 layers and all 32 Llama-2-7B layers across the three forget splits. Full-model SatImp provides the baseline in each setting. The green shaded regions mark layers with both lower forget ES and higher retain ES than SatImp. Stars and hollow orange diamonds identify the FS- and BLS-selected layers reported in Table 1, respectively.

Across the six TOFU settings, 80 of 168 layer runs improve both metrics over SatImp. These include many layers beyond those selected by FS and BLS, with effective layers near the beginning and end of each model. To summarize this pattern, we group layers by relative depth into early, middle, and late thirds. Pooling the five highest-NFR layers from each of the six settings gives 30 selections: 17 early and 13 late, with none in the middle group. Phi-1.5 contributes seven early and eight late selections, while Llama-2-7B contributes ten early and five late selections. These results show that CoLU is effective at different depths, with the strongest results concentrated near either end of the model. Performance still varies substantially across layers and forget splits, so selecting among these effective layers remains important for achieving strong forgetting and retention.

6.3 Layer Selection Methods

To determine how best to select among these layers, we compare BLS with gradient-based selection scores adapted from SLUG (Cai et al., 2025), FOM-UL (Ranjan et al., 2026), and SURE (Zhang et al., 2025). Each selector shortlists three layers for each TOFU setting. We perform full-data unlearning separately for each candidate and evaluate selection quality against the corresponding

full-screening results. The gradient scores are computed at the original checkpoint: FOM-UL ratio compares forget and retain gradient norms, SLUG importance normalizes the forget gradient norm by the layer's parameter norm, and SURE-inspired selection uses the forget gradient norm alone. We apply these scores to complete transformer layers with a fixed shortlist size. This comparison evaluates their ability to select layers for CoLU; it does not reproduce the full unlearning procedures of the original methods.

For each setting, the targets are the three highest-NFR layers in the full sweep. *Shortlist hit rate* is the percentage of settings in which a shortlist contains at least one target. *Recall@3* is the percentage of the three targets recovered, averaged across settings. *Winner NFR* is the highest full-data NFR score among the three shortlisted layers. Hit rate and recall measure recovery of the top three layers, while winner NFR measures the quality of the selected model.

Table 4: Comparison of layer selection methods on TOFU. Higher values are better for all three metrics. Best values per row are bold, including ties. Random-three reports expected values for uniform sampling of three distinct layers.

Model	Split	Shortlist hit rate (%) / Recall@3 (%) / Winner NFR ↑				
		BLS (ours)	FOM-UL ratio	SLUG importance	SURE-inspired	Random-three (expected)
Phi-1.5	Forget-1%	**100/66.7/94.5**	**100/66.7/94.5**	0/0/91.3	0/0/91.3	34.3/12.5/92.4
	Forget-5%	**100/33.3/90.4**	0/0/89.1	0/0/89.4	0/0/89.4	34.3/12.5/89.8
	Forget-10%	**100/66.7**/91.5	**100**/33.3/91.5	**100**/33.3/**92.9**	**100**/33.3/**92.9**	34.3/12.5/88.7
	Average	**100/55.6/92.2**	66.7/33.3/91.7	33.3/11.1/91.2	33.3/11.1/91.2	34.3/12.5/90.3
Llama-2-7B	Forget-1%	**100/33.3/98.1**	**100/33.3**/97.3	0/0/92.0	0/0/92.0	26.3/9.4/95.5
	Forget-5%	**100/33.3/97.6**	**100/33.3**/97.4	0/0/95.8	0/0/96.0	26.3/9.4/96.7
	Forget-10%	**100/66.7/98.5**	0/0/94.3	**100**/33.3/98.3	0/0/96.8	26.3/9.4/96.3
	Average	**100/44.4/98.1**	66.7/22.2/96.3	33.3/11.1/95.4	0/0/94.9	26.3/9.4/96.1
Both models	Overall	**100/50.0/95.1**	66.7/27.8/94.0	33.3/11.1/93.3	16.7/5.6/93.1	30.3/10.9/93.2

Table 4 shows that BLS achieves a 100% shortlist hit rate across all six settings, compared with 66.7% for FOM-UL ratio, 33.3% for SLUG importance, and 16.7% for SURE-inspired selection. BLS recovers nine of the 18 target positions, giving an overall Recall@3 of 50.0%, compared with five for FOM-UL ratio (27.8%). It matches or exceeds every comparator on both metrics in each setting. BLS also achieves the highest or tied-highest winner NFR in five of six settings and the highest average, at 95.11. The exception is Phi-1.5 Forget-10%, where SLUG importance and SURE-inspired selection obtain higher winner NFR. Compared with Random-three, BLS improves winner NFR in all six settings, averaging 95.11 versus an expected 93.22, a gain of 1.89 points. For context, the gap between the best and worst layers in the full sweep averages 12.08 NFR points across the six settings. Appendix B reports these ranges and rescales winner NFR to each setting's observed range. On this scale, BLS averages 94.8, compared with 77.6 for Random-three.

6.4 Runtime and Memory Cost

Table 5: Total training time on TOFU (min). NPO and SatImp update the full model. CoLU (BLS) includes screening and training the three selected layers; CoLU (FS) sums training across all layers.

Model	Split	NPO	SatImp	CoLU (BLS)	Runtime vs. SatImp	CoLU (FS)	Runtime vs. SatImp
Phi-1.5	Forget-1%	2.69	1.48	2.90	1.96×	7.29	4.93×
	Forget-5%	13.09	4.62	9.89	2.14×	33.88	7.33×
	Forget-10%	25.54	10.60	19.05	1.80×	66.93	6.31×
Llama-2-7B	Forget-1%	7.18	4.83	7.68	1.59×	25.01	5.18×
	Forget-5%	35.65	26.28	24.09	0.92×	118.44	4.51×
	Forget-10%	70.78	44.18	52.86	1.20×	234.05	5.30×

Table 5 compares total training time for full-model NPO and SatImp with CoLU (FS) and CoLU (BLS). For Phi-1.5 on the Forget-5% split, CoLU with BLS takes 9.89 min, compared with

33.88 min for a full sweep and 4.62 min for full-model SatImp. The BLS total is $2.14\times$ the SatImp training time. Averaging the training-time ratio across the three forget splits gives $1.97\times$ for Phi-1.5 and $1.23\times$ for Llama-2-7B. Relative to FS, BLS reduces total training time by 60.2–79.7% across the six settings. These totals include brief screening and full-data training of the three candidates.

Table 6: GPU memory usage (GiB). TOFU use CWA method as baseline. Single-layer means average memory usage across runs for all transformer layers.

Dataset/model	Full model	Single-layer mean
TOFU Phi-1.5	33.33	7.53
TOFU Llama-2-7B	100.41	21.91
MUSE Books	113.30	23.18
MUSE News	109.30	22.78

Speaking of the memory cost, restricting updates to one layer reduces the memory needed for gradients and optimizer states sine one layer in Llama-2-7B only takes approximately 3% of the model's parameters. Table 6 compares full-model unlearning memory cost with the mean memory usage of single-layer runs across all transformer layers. Across the four settings, CoLU reduces average GPU memory usage by 77.4% to 79.5%. On TOFU, usage falls from 33.33 to 7.53 GiB for Phi-1.5 and from 100.41 to 21.91 GiB for Llama-2-7B. On MUSE Books and News, the single-layer averages are 23.18 and 22.78 GiB, compared with 113.30 and 109.30 GiB for full-model unlearning, respectively. These savings make CoLU more practical when GPU memory limits the use of full-model unlearning.

7 Discussion and Limitations

The experiments above show that CoLU can improve both forgetting and retention over several full-model and multi-layer baselines significantly. BLS reduces the cost of a full sweep, but CoLU can still take longer than full-model unlearning. However, it requires substantially less GPU memory compared to full-model unlearning which make it suitable for many resource-constrained environments. The adapted gradient-based scores from previous work are less effective at identifying promising CoLU layers than BLS, indicating that these criteria do not consistently predict performance after unlearning. Future work should therefore explore selection methods that improve on both BLS and existing criteria, reducing screening cost while reliably identifying effective layers across models and forget splits.

8 Conclusion

We presented CoLU, which combines confidence-weighted ascent with unlearning in a single selected transformer layer. CoLU improves both forgetting and retention over SOTAs on TOFU and achieves the highest retained KnowMem and the PrivLeak closest to zero among the compared unlearning methods on MUSE Books and News. BLS provides results comparable to full screening on TOFU at substantially lower training cost. Single-layer updates also reduce GPU memory usage, though screening remains a source of runtime overhead.

AI-Use Statement

The authors conceived the research idea and developed the method. We used OpenAI Codex for grammar correction, language editing, table formatting, and for coding assistance when reproducing related methods for our ablation study. We also used Codex to compute summary statistics and check numerical consistency. We used ChatGPT for pre-submission manuscript review, including feedback on methodology and experiments. The authors take responsibility for the final manuscript, code, and reported results.

REPRODUCIBILITY STATEMENT

The method section describes the unlearning objective and layer-selection procedures. The appendix provides implementation details, hyperparameters, training schedules, and additional results across three random seeds for Phi-1.5 on TOFU. We plan to release the implementation upon acceptance.

REFERENCES

Saransh Agrawal and Kuan-Hao Huang. Sha256 at semeval-2025 task 4: Selective amnesia–constrained unlearning for large language models via knowledge isolation. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pp. 2368–2375, 2025.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *Proceedings - 2021 IEEE Symposium on Security and Privacy, SP 2021*, Proceedings - IEEE Symposium on Security and Privacy, pp. 141–159, United States, May 2021. Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/SP40001.2021.00019.

Zikui Cai, Yaoteng Tan, and M. Salman Asif. Targeted unlearning with single layer unlearning gradient. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/cai25f.html.

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

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 Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/02443e4e008231e0af0855f3cc70ed17-Abstract-Conference.html.

Jiahui Geng, Qing Li, Herbert Woisetschlaeger, Zongxiong Chen, Fengyu Cai, Yuxia Wang, Preslav Nakov, Hans-Arno Jacobsen, and Fakhri Karray. A comprehensive survey of machine unlearning techniques for large language models. *arXiv preprint arXiv:2503.01854*, 2025.

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 Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 14389–14408. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.805. URL https://doi.org/10.18653/v1/2023.acl-long.805.

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 1957–1987. Curran Associates, Inc., 2023. doi: 10.52202/075280-0095. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/062d711fb777322e2152435459e6e9d9-Paper-Conference.pdf.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, et al. Rethinking machine unlearning for large language models. *Nature Machine Intelligence*, 7(2):181–194, 2025.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. TOFU: A task of fictitious unlearning for llms. *CoRR*, abs/2401.06121, 2024. doi: 10.48550/ARXIV.2401.06121. URL https://doi.org/10.48550/arXiv.2401.06121.

Milad Nasr, Javier Rando, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher Choquette-Choo, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from aligned, production language models. In *International Conference on Learning Representations*, volume 2025, pp. 82363–82435, 2025.

Vaidehi Patil, Peter Hase, and Mohit Bansal. Can sensitive information be deleted from llms? objectives for defending against extraction attacks. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=7erlRDoaV8.

Ravi Ranjan, Olivera Kotevska, and Agoritsa Polyzou. Forgetting Only What Matters: Layer-Selective Unlearning toward Robust LLMs. *arXiv preprint arXiv:2609.10439*, 2026. URL https://arxiv.org/abs/2609.10439.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A. Smith, and Chiyuan Zhang. MUSE: machine unlearning six-way evaluation for language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=TArmA033BU.

Qizhou Wang, Bo Han, Puning Yang, Jianing Zhu, Tongliang Liu, and Masashi Sugiyama. Towards effective evaluations and comparisons for LLM unlearning methods. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025a. URL https://openreview.net/forum?id=wUtCieKuQU.

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, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025b. URL https://openreview.net/forum?id=huo8MqVH6t.

Puning Yang, Qizhou Wang, Zhuo Huang, Tongliang Liu, Chengqi Zhang, and Bo Han. Exploring criteria of loss reweighting to enhance llm unlearning. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML'25. JMLR.org, 2025.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 105425–105475. Curran Associates, Inc., 2024. doi: 10.52202/079017-3346. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/be52acf6bccf4a8c0a90fe2f5cfcead3-Paper-Conference.pdf.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. *CoRR*, abs/2404.05868, 2024. doi: 10.48550/ARXIV.2404.05868. URL https://doi.org/10.48550/arXiv.2404.05868.

Zhiwei Zhang, Fali Wang, Xiaomin Li, Zongyu Wu, Xianfeng Tang, Hui Liu, Qi He, Wenpeng Yin, and Suhang Wang. Catastrophic failure of LLM unlearning via quantization. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=lHSeDYamnz.

A Hyperparameters and Selection Rules

Shared CoLU settings. All CoLU runs use a batch size of 16. On both TOFU and MUSE, BLS trains each layer for two epochs on a fixed 20% subset of the training data, then selects $K = 3$ candidates for full-data unlearning. Within each experimental setting, screening uses the same optimizer, batch size, and loss hyperparameters as full-data training, with a higher learning rate. We hold the screening learning rate fixed across all layers. The full-data learning rates below apply to both FS and the final candidate runs in BLS.

TOFU configurations. Full-data runs train one transformer layer for five epochs using paged AdamW with 32-bit optimizer states and zero weight decay. We set $A = 0.65$, $\lambda_f = 1$, and $\lambda_r = 0.1$ for all TOFU settings. Table 7 lists the learning rates, A and α values, and loss weights for each model and forget split. Screening and full-data training use the same α value.

Table 7: CoLU hyperparameters on TOFU. The BLS learning rate applies to the brief screening trials.

Model	Forget split	Full-data LR	BLS LR	A	α	λ_f	λ_r
Phi-1.5	1%	2×10^{-4}	4×10^{-4}	0.65	6.0	1	0.1
	5%	2×10^{-4}	6×10^{-4}	0.65	4.5	1	0.1
	10%	2×10^{-4}	6×10^{-4}	0.65	4.5	1	0.1
Llama-2-7B	1%	1×10^{-4}	3×10^{-4}	0.65	4.5	1	0.1
	5%	1×10^{-4}	4×10^{-4}	0.65	4.5	1	0.1
	10%	1×10^{-4}	3×10^{-4}	0.65	4.5	1	0.1

MUSE configurations. On both Books and News, FS trains each layer for eight epochs on the full data. We select the best checkpoint across epochs, following prior MUSE evaluations (Fan et al., 2025; Yang et al., 2025). BLS screens layers for two epochs on a fixed 20% subset, then restarts each shortlisted candidate from the original checkpoint and trains it for eight epochs on the full data. We use a full-data learning rate of 1×10^{-4} and a BLS screening learning rate of 3×10^{-4} in both domains. Both domains use AdamW (`adamw_torch`) with zero weight decay and $\lambda_f = 1$. We set $\lambda_r = 1$ for Books and $\lambda_r = 0.4$ for News. Table 8 summarizes these settings and the loss hyperparameters, which are shared by screening and full-data runs.

Table 8: CoLU hyperparameters on MUSE. The BLS learning rate applies to the two-epoch screening trials.

Domain	Full-data LR	BLS LR	A	α	λ_f	λ_r
Books	1×10^{-4}	3×10^{-4}	1	4.6	1	1
News	1×10^{-4}	3×10^{-4}	1	3.0	1	0.4

B Range-Normalized Layer-Selection Scores

All Phi-1.5 NFR calculations use the fixed reference $R_O = 0.6710792227$ for all three splits and $F_O = 0.6262454642$, 0.6990105329, and 0.7041712797 for Forget-1%, Forget-5%, and Forget-10%, respectively.

To put winner-score differences in the context of the variation across layers, we rescale NFR separately for each model and forget split. For setting t, let $s_{\min,t}$ and $s_{\max,t}$ be the lowest and highest NFR scores in the full layer sweep, and let $s_{a,t}$ be the winner NFR for selector a. We report

$$z_{a,t} = 100 \, \frac{s_{a,t} - s_{\min,t}}{s_{\max,t} - s_{\min,t}}. \tag{4}$$

A score of 0 corresponds to the worst layer and 100 to the best layer within that setting. We average $z_{a,t}$ across settings after rescaling. This supplementary analysis describes relative selection

quality within each sweep; it leaves the NFR selection rule unchanged. For Random-three, the same transformation applies to the expected winner NFR because the bounds are fixed within each setting.

Table 9: Observed layer NFR bounds for the supplementary comparison.

Model	Split	Minimum (layer)	Maximum (layer)	Range width
Phi-1.5	Forget-1%	87.38 (L8)	94.55 (L20)	7.17
	Forget-5%	82.46 (L21)	92.22 (L4)	9.76
	Forget-10%	81.94 (L8)	92.92 (L3)	10.98
Llama-2-7B	Forget-1%	73.16 (L31)	98.09 (L4)	24.92
	Forget-5%	89.62 (L0)	97.64 (L3)	8.02
	Forget-10%	86.85 (L31)	98.50 (L3)	11.64
Both models	Average			12.08

Table 10: Range-normalized winner NFR (↑) from Eq. (4). Best values per row are bold, including ties.

Model	Split	BLS (ours)	FOM-UL ratio	SLUG importance	SURE-inspired	Random-three (expected)
Phi-1.5	Forget-1%	**100.0**	**100.0**	55.1	55.1	70.5
	Forget-5%	**81.2**	67.8	71.4	71.4	75.1
	Forget-10%	87.3	87.3	**100.0**	**100.0**	61.2
	Average	**89.5**	85.0	75.5	75.5	69.0
Llama-2-7B	Forget-1%	**100.0**	96.7	75.4	75.4	89.6
	Forget-5%	**100.0**	97.1	77.1	79.1	88.1
	Forget-10%	**100.0**	63.8	98.6	85.1	80.9
	Average	**100.0**	85.8	83.7	79.8	86.2
Both models	Overall	**94.8**	85.4	79.6	77.7	77.6

Under the bounds in Table 9, layer NFR ranges span 7.17–24.92 points, with a mean width of 12.08 points. Table 10 places each selector's winner within these ranges. BLS averages 94.8, compared with 77.6 for Random-three; these are positions on a rescaled score, rather than percentages of information forgotten.

C Effect of Retain Regularization

At the existing learning rates and training budgets, removing retain regularization substantially reduces retention. On Forget-5% and Forget-10%, even full layer search cannot recover a useful forget–retain trade-off. These results support the importance of retain regularization in the current CoLU configuration.

D Variation Across Training Seeds

We evaluate Phi-1.5 with training seeds 1, 2, and 3 on all three TOFU forget splits, using the configurations in Appendix A. Each seed includes a full-data run for every layer and a separate set of BLS screening trials. FS selects the highest-NFR result among all 24 layers; BLS selects the highest-NFR result within its three-layer screening shortlist. For this analysis, BLS uses the corresponding full-sweep checkpoints, so FS and BLS share the same evaluation when they select the same layer. The screening subsets remain fixed across training seeds.

Table 12 reports the mean and sample standard deviation of the three selected outcomes for each method and split. Table 13 instead summarizes each fixed layer across seeds. FQ and MU are reevaluated under a common protocol for all three seeds; these evaluations differ from the historical

Table 11: Retain-regularization ablation for CoLU (FS) on Phi-1.5 across TOFU splits. We report the result with the highest NFR score from each 24-layer sweep, without a baseline-dominance filter.

Method	Split	Layer	ES Forget ↓	ES Retain ↑
With retain regularization				
CoLU (FS)	Forget-1%	20	0.035851	0.636313
CoLU (FS)	Forget-5%	4	0.020209	0.586068
CoLU (FS)	Forget-10%	3	0.003480	0.579316
Without retain regularization				
CoLU (FS)	Forget-1%	20	0.024284	0.458252
CoLU (FS)	Forget-5%	1†	0.000000	0.000000
CoLU (FS)	Forget-10%	0	0.000000	0.000192

†Layers 1–23 tie with zero Forget and Retain ES; layer 1 is reported by the layer-index tie-break rule. This tie reflects retention collapse, not successful unlearning.

evaluations in Table 1. Neither FQ nor MU is used for selection. FQ means and standard deviations are descriptive summaries of the individual test p-values, not a pooled test.

Seed 1 reuses historical training runs and ES measurements, while seeds 2 and 3 use the multi-seed training launcher. The observed dispersion may therefore include execution differences in addition to training-seed variation. It does not measure variation from resampling the screening subsets. ES is used for both selection and reporting, so the selected scores are not independent held-out estimates.

Table 12: Selected CoLU results on Phi-1.5 across three training seeds. Each metric cell gives the mean above and ± sample standard deviation below ($n = 3$). Layer indices are zero-based and listed in seed order 1/2/3. FS selects the highest-NFR layer in each full sweep; BLS selects the highest-NFR layer within its screening shortlist.

Method	Layers (1/2/3)	Forget ES ↓	Retain ES ↑	NFR ↑	FQ ↑	MU ↑
Forget-1%						
CoLU (FS)	20/20/20	0.0337 ± 0.0019	0.6406 ± 0.0048	95.04 ± 0.47	0.2079 ± 0.1791	0.5035 ± 0.0021
CoLU (BLS)	20/20/20	0.0337 ± 0.0019	0.6406 ± 0.0048	95.04 ± 0.47	0.2079 ± 0.1791	0.5035 ± 0.0021
Forget-5%						
CoLU (FS)	4/1/2	0.0115 ± 0.0101	0.5712 ± 0.0202	91.74 ± 0.80	0.0152 ± 0.0213	0.5030 ± 0.0058
CoLU (BLS)	23/18/2	0.0097 ± 0.0102	0.5473 ± 0.0029	90.08 ± 0.92	0.1167 ± 0.1834	0.4951 ± 0.0228
Forget-10%						
CoLU (FS)	3/1/2	0.0035 ± 0.0011	0.5741 ± 0.0095	92.53 ± 0.78	2.97×10^{-8} $\pm 5.08\times10^{-8}$	0.5025 ± 0.0031
CoLU (BLS)	2/2/2	0.0023 ± 0.0022	0.5614 ± 0.0017	91.66 ± 0.16	4.96×10^{-7} $\pm 7.85\times10^{-7}$	0.4977 ± 0.0071

FS and BLS select layer 20 in all three Forget-1% runs. The selected layers vary on Forget-5%, while BLS selects layer 2 in all three Forget-10% runs. On Forget-5% and Forget-10%, BLS has lower mean forget ES than FS, accompanied by lower mean retain ES and NFR.

Table 13: Full-data CoLU results for every Phi-1.5 layer across three training seeds. Values are mean ± sample standard deviation ($n = 3$), computed for the same layer across seeds. Each split contains 24 layers, covering 216 full-data runs.

Layer	Forget ES ↓	Retain ES ↑	NFR ↑	FQ ↑	MU ↑
Forget-1%					
0	0.0377 ± 0.0099	0.6012 ± 0.0057	91.78 ± 0.83	0.8916 ± 0.1145	0.4970 ± 0.0035
1	0.0428 ± 0.0111	0.5829 ± 0.0194	90.02 ± 2.31	0.4743 ± 0.1807	0.5025 ± 0.0067
2	0.0300 ± 0.0058	0.5977 ± 0.0073	92.14 ± 0.97	0.8169 ± 0.0883	0.5012 ± 0.0035
3	0.0349 ± 0.0008	0.5785 ± 0.0127	90.32 ± 1.00	0.4626 ± 0.1005	0.5058 ± 0.0054
4	0.0328 ± 0.0028	0.5896 ± 0.0107	91.31 ± 1.00	0.4163 ± 0.1568	0.5013 ± 0.0034
5	0.0299 ± 0.0054	0.5678 ± 0.0182	89.92 ± 0.96	0.4046 ± 0.0000	0.5039 ± 0.0025
6	0.0418 ± 0.0043	0.5815 ± 0.0138	89.99 ± 1.00	0.1054 ± 0.0559	0.5052 ± 0.0043
7	0.0377 ± 0.0043	0.5718 ± 0.0166	89.60 ± 1.42	0.0828 ± 0.0248	0.5058 ± 0.0014
8	0.0374 ± 0.0037	0.5528 ± 0.0138	88.20 ± 0.87	0.0828 ± 0.0248	0.5013 ± 0.0028
9	0.0353 ± 0.0042	0.5464 ± 0.0062	87.89 ± 0.38	0.0456 ± 0.0147	0.4981 ± 0.0027
10	0.0289 ± 0.0021	0.5552 ± 0.0061	89.06 ± 0.41	0.1985 ± 0.0581	0.5032 ± 0.0068
11	0.0297 ± 0.0033	0.5510 ± 0.0123	88.69 ± 1.07	0.1197 ± 0.0392	0.4978 ± 0.0046
12	0.0274 ± 0.0035	0.5567 ± 0.0073	89.29 ± 0.73	0.0971 ± 0.0000	0.5071 ± 0.0045
13	0.0240 ± 0.0021	0.5601 ± 0.0156	89.81 ± 1.21	0.1423 ± 0.0392	0.5050 ± 0.0034
14	0.0250 ± 0.0062	0.5786 ± 0.0143	91.11 ± 0.82	0.1423 ± 0.0392	0.5062 ± 0.0056
15	0.0253 ± 0.0008	0.5601 ± 0.0139	89.71 ± 0.99	0.1533 ± 0.0973	0.5041 ± 0.0031
16	0.0200 ± 0.0017	0.5627 ± 0.0054	90.33 ± 0.48	0.2321 ± 0.0581	0.5060 ± 0.0011
17	0.0209 ± 0.0032	0.5951 ± 0.0159	92.67 ± 0.94	0.3120 ± 0.0802	0.4986 ± 0.0026
18	0.0291 ± 0.0024	0.6118 ± 0.0124	93.26 ± 1.06	0.2784 ± 0.1203	0.5030 ± 0.0044
19	0.0328 ± 0.0026	0.6135 ± 0.0129	93.09 ± 0.77	0.2448 ± 0.1383	0.4969 ± 0.0030
20	0.0337 ± 0.0019	0.6406 ± 0.0048	95.04 ± 0.47	0.2079 ± 0.1791	0.5035 ± 0.0021
21	0.0424 ± 0.0050	0.6165 ± 0.0079	92.55 ± 0.94	0.2222 ± 0.1615	0.5036 ± 0.0010
22	0.0476 ± 0.0044	0.5881 ± 0.0175	90.01 ± 1.20	0.7544 ± 0.1704	0.4901 ± 0.0036
23	0.0804 ± 0.0099	0.5988 ± 0.0075	88.19 ± 0.39	0.1197 ± 0.0392	0.4972 ± 0.0042

Continued on next page.

Table 13: Full-data CoLU results for every Phi-1.5 layer (continued).

Layer	Forget ES ↓	Retain ES ↑	NFR ↑	FQ ↑	MU ↑
Forget-5%					
0	0.0073 ± 0.0047	0.5340 ± 0.0222	89.26 ± 1.50	$(5.52 \pm 5.94) \times 10^{-5}$	0.4835 ± 0.0062
1	0.0129 ± 0.0076	0.5572 ± 0.0193	90.59 ± 1.50	0.0494 ± 0.0803	0.5033 ± 0.0057
2	0.0046 ± 0.0059	0.5494 ± 0.0066	90.61 ± 0.84	0.0054 ± 0.0094	0.4997 ± 0.0051
3	0.0131 ± 0.0056	0.5531 ± 0.0176	90.27 ± 1.12	0.1015 ± 0.1469	0.4957 ± 0.0073
4	0.0180 ± 0.0046	0.5720 ± 0.0169	91.33 ± 0.96	0.3116 ± 0.4797	0.5038 ± 0.0041
5	0.0221 ± 0.0055	0.5456 ± 0.0145	89.07 ± 1.07	0.0095 ± 0.0059	0.4926 ± 0.0074
6	0.0236 ± 0.0027	0.5374 ± 0.0041	88.35 ± 0.49	0.0017 ± 0.0012	0.4997 ± 0.0040
7	0.0194 ± 0.0033	0.5301 ± 0.0152	88.10 ± 1.28	0.0015 ± 0.0014	0.4981 ± 0.0051
8	0.0216 ± 0.0048	0.5296 ± 0.0125	87.91 ± 1.15	$(3.39 \pm 2.68) \times 10^{-4}$	0.4956 ± 0.0051
9	0.0230 ± 0.0032	0.5226 ± 0.0169	87.29 ± 1.43	$(1.02 \pm 0.721) \times 10^{-4}$	0.4938 ± 0.0038
10	0.0214 ± 0.0039	0.5186 ± 0.0099	87.10 ± 0.63	$(9.36 \pm 10.3) \times 10^{-4}$	0.4941 ± 0.0052
11	0.0164 ± 0.0041	0.5127 ± 0.0081	87.03 ± 0.31	0.0026 ± 0.0015	0.4863 ± 0.0007
12	0.0186 ± 0.0014	0.5205 ± 0.0084	87.45 ± 0.66	$(4.21 \pm 2.33) \times 10^{-4}$	0.4829 ± 0.0053
13	0.0240 ± 0.0023	0.5135 ± 0.0140	86.55 ± 1.13	$(6.1 \pm 3.09) \times 10^{-4}$	0.4907 ± 0.0126
14	0.0265 ± 0.0024	0.5349 ± 0.0026	87.96 ± 0.12	$(3.64 \pm 3.24) \times 10^{-4}$	0.4856 ± 0.0082
15	0.0307 ± 0.0028	0.5410 ± 0.0087	88.11 ± 0.53	$(1.73 \pm 0.953) \times 10^{-4}$	0.4912 ± 0.0020
16	0.0233 ± 0.0037	0.5527 ± 0.0109	89.51 ± 0.72	0.0051 ± 0.0060	0.4945 ± 0.0030
17	0.0445 ± 0.0065	0.5295 ± 0.0060	86.27 ± 0.55	0.0056 ± 0.0033	0.4955 ± 0.0058
18	0.0203 ± 0.0010	0.5441 ± 0.0013	89.09 ± 0.04	0.1452 ± 0.1590	0.5091 ± 0.0032
19	0.0417 ± 0.0101	0.5338 ± 0.0288	86.78 ± 1.42	0.0267 ± 0.0221	0.4967 ± 0.0041
20	0.0242 ± 0.0035	0.5434 ± 0.0168	88.75 ± 1.14	0.0705 ± 0.0622	0.4965 ± 0.0007
21	0.0604 ± 0.0147	0.5394 ± 0.0380	85.87 ± 3.73	0.0212 ± 0.0147	0.4940 ± 0.0025
22	0.0686 ± 0.0021	0.5669 ± 0.0172	87.33 ± 1.39	0.0032 ± 0.0025	0.4910 ± 0.0095
23	0.0090 ± 0.0010	0.5513 ± 0.0021	90.43 ± 0.11	0.0220 ± 0.0176	0.4745 ± 0.0051

Continued on next page.

Table 13: Full-data CoLU results for every Phi-1.5 layer (continued).

Layer	Forget ES ↓	Retain ES ↑	NFR ↑	FQ ↑	MU ↑
Forget-10%					
0	0.0156 ± 0.0125	0.5284 ± 0.0228	88.26 ± 2.17	0.0489 ± 0.0846	0.4951 ± 0.0108
1	0.0108 ± 0.0117	0.5447 ± 0.0326	89.82 ± 2.79	0.0010 ± 0.0018	0.4972 ± 0.0041
2	0.0023 ± 0.0022	0.5614 ± 0.0017	91.66 ± 0.16	$(4.96 \pm 7.85) \times 10^{-7}$	0.4977 ± 0.0071
3	0.0035 ± 0.0015	0.5632 ± 0.0157	91.71 ± 1.22	$(6.77 \pm 5.66) \times 10^{-9}$	0.4983 ± 0.0083
4	0.0097 ± 0.0070	0.5115 ± 0.0249	87.42 ± 2.18	0.0069 ± 0.0120	0.5007 ± 0.0104
5	0.0033 ± 0.0026	0.5327 ± 0.0340	89.45 ± 2.70	$(2.95 \pm 5.1) \times 10^{-8}$	0.4978 ± 0.0048
6	0.0102 ± 0.0028	0.4998 ± 0.0186	86.51 ± 1.57	0.0219 ± 0.0380	0.5002 ± 0.0061
7	0.0076 ± 0.0027	0.4670 ± 0.0092	84.25 ± 0.78	0.0011 ± 0.0012	0.4898 ± 0.0086
8	0.0021 ± 0.0015	0.4456 ± 0.0117	83.04 ± 0.96	$(5.72 \pm 6.29) \times 10^{-4}$	0.4964 ± 0.0050
9	0.0078 ± 0.0008	0.4859 ± 0.0049	85.65 ± 0.38	0.0080 ± 0.0072	0.4997 ± 0.0066
10	0.0045 ± 0.0020	0.4885 ± 0.0180	86.08 ± 1.31	$(1.45 \pm 1.28) \times 10^{-5}$	0.5023 ± 0.0032
11	0.0056 ± 0.0016	0.5238 ± 0.0217	88.63 ± 1.51	$(2 \pm 2.7) \times 10^{-6}$	0.4990 ± 0.0073
12	0.0061 ± 0.0014	0.4956 ± 0.0152	86.49 ± 1.19	0.0195 ± 0.0213	0.4995 ± 0.0046
13	0.0077 ± 0.0020	0.4692 ± 0.0086	84.41 ± 0.55	0.0016 ± 0.0022	0.4881 ± 0.0028
14	0.0275 ± 0.0088	0.5098 ± 0.0117	86.03 ± 0.77	$(2.67 \pm 1.75) \times 10^{-4}$	0.4956 ± 0.0058
15	0.0534 ± 0.0018	0.5361 ± 0.0049	86.15 ± 0.24	$(7.8 \pm 3.92) \times 10^{-5}$	0.5013 ± 0.0008
16	0.0452 ± 0.0085	0.5392 ± 0.0053	86.96 ± 0.25	$(2.69 \pm 0.566) \times 10^{-4}$	0.4973 ± 0.0036
17	0.0696 ± 0.0074	0.5491 ± 0.0084	85.97 ± 0.33	$(8.29 \pm 4.61) \times 10^{-4}$	0.5038 ± 0.0026
18	0.0425 ± 0.0061	0.5505 ± 0.0069	88.00 ± 0.95	0.0042 ± 0.0047	0.5041 ± 0.0028
19	0.0568 ± 0.0121	0.5750 ± 0.0050	88.81 ± 0.93	$(7.79 \pm 8.14) \times 10^{-4}$	0.4986 ± 0.0017
20	0.0544 ± 0.0111	0.5618 ± 0.0073	88.00 ± 0.53	$(5.25 \pm 3.3) \times 10^{-4}$	0.5009 ± 0.0022
21	0.0719 ± 0.0253	0.5258 ± 0.0117	84.07 ± 2.66	0.0029 ± 0.0038	0.4910 ± 0.0058
22	0.0601 ± 0.0068	0.5446 ± 0.0071	86.30 ± 0.84	$(4.92 \pm 3.62) \times 10^{-4}$	0.4952 ± 0.0011
23	0.0079 ± 0.0044	0.5205 ± 0.0085	88.21 ± 0.33	$(4.46 \pm 6.93) \times 10^{-4}$	0.4785 ± 0.0097