CLUE: Closing the Loop on Conflict and Collapse in LLM Unlearning

Mengyang Li[1] **Jingwen Wang**[1] **Yu Zhang**[1] **Pinlong Zhao**[2*]
[1]Tianjin Key Laboratory of Wireless Mobile Communications and Power Transmission, Tianjin Normal University, Tianjin, China
[2]School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China
pinlongzhao@hdu.edu.cn

Abstract

Machine unlearning for large language models optimizes two opposing losses: a forget objective drives the model away from targeted knowledge, while a retain objective holds it near its original behavior. Existing methods commit to a fixed schedule of learning rate, retain weight, and step count chosen by offline grid search, which is blind to how the trajectory actually evolves. We show that LLM unlearning trajectories share a three-phase structure across all seven unlearning methods we evaluate: an early phase where the two gradients are nearly orthogonal, a conflict phase where they oppose each other and retain loss begins to grow, and a terminal phase where representational divergence crosses an empirically calibrated threshold and retain loss no longer recovers. Each phase calls for a different control policy. We propose CLUE, a closed-loop framework whose architectural choices each correspond to a phase or transition: a dual-tower controller separates forget and retain signal pathways, a conflict-driven gate switches between them, and an explicit collapse-risk head supervises anticipatory stopping. The controller is trained by task-distribution meta-optimization. We provide a single-step bound on conflict-induced retain loss growth, a finite-horizon recovery bound, and a stopping regret bound. On TOFU, WMDP, and MUSE with Llama-2/3 and Mistral models from 7B to 70B, CLUE achieves better forget-quality versus model-utility trade-offs than fixed-strategy baselines, stops within 5% of the oracle, and remains robust under adversarial prompting, relearning, and INT4 quantization.

1 Introduction

Large language models trained on web-scale corpora inevitably memorize sensitive, private, or copyrighted information [Carlini et al., 2021, 2023]. Machine unlearning has emerged as a practical alternative to costly full retraining for selectively removing the influence of specific training data [Cao and Yang, 2015, Bourtoule et al., 2021], with applications spanning privacy compliance under regulations such as GDPR and CCPA, hazardous-knowledge removal [Li et al., 2024], and copyright takedowns [Eldan and Russinovich, 2023].

We begin not with a method but with a phenomenon. Across every LLM unlearning method we examined, namely GA [Yao et al., 2024], GradDiff [Yao et al., 2024], KL [Maini et al., 2024], NPO [Zhang et al., 2024], SimNPO [Fan et al., 2025], RMU [Li et al., 2024], and task arithmetic [Ilharco et al., 2023], the training trajectory traverses three structurally distinct phases. In an *early* phase, the forget loss rises while the retain loss stays flat and the cosine similarity between the forget and retain gradients, $\cos(g_f, g_r)$ ("cosine" below), hovers near zero. In a *conflict* phase, the cosine turns sharply negative, mid-layer representations of the forget and retain sets begin to diverge, and the retain loss

*Corresponding author.

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

starts to grow. In a *terminal* phase, the logit margin on forget probes collapses toward zero and, in all our runs, retain loss does not recover under any subsequent step (Section 4.3); a finite-horizon analysis explains why bounded corrective steps cannot undo damage accumulated over a long conflict phase (Proposition B.2). The phase boundaries are sharp, shared by all evaluated methods with the same thresholds, and observable in real time from quantities computed during ordinary backward passes. Each phase calls for a different control policy, and no fixed schedule can satisfy all three. The literature has named the symptoms of fixed schedules [Maini et al., 2024, Zhang et al., 2024], including catastrophic collapse [Zhang et al., 2024], over-forgetting [Yuan et al., 2025b, Wang et al., 2025], and representational erasure [Li et al., 2024, Xu et al., 2026], but has not, to our knowledge, identified the underlying phase structure as a unified explanation.

We propose CLUE, a closed-loop framework whose architectural choices each correspond to a phase or transition. CLUE extracts ten diagnostic signals per step organized into forget, retain, and conflict groups, two of which are specific to unlearning: a logit-margin collapse indicator and a mid-layer forget-retain representation divergence. A dual-tower controller processes forget and retain signals through separate recurrent pathways and fuses them through a gate driven by the conflict group, so that conflict modulates how the two objectives interact. Four heads produce step-wise decisions on the unlearning rate, the retain weight, a continuation probability, and an explicit collapse-risk estimate; stopping fires whenever risk exceeds threshold or continuation probability drops, turning stopping from a reactive convergence criterion into an anticipatory control action. The controller is trained by task-distribution meta-optimization, with each update evaluated on question-answer pairs that the inner loop never sees. Existing training controllers [Li et al., 2026c, Ren et al., 2018, Shu et al., 2019] operate where conflict is small; CLUE makes conflict a first-class observable, which the high-conflict regime of unlearning requires (Section 3.1). We report TOFU forget quality (FQ, the KS-test p-value between the truth-ratio distributions of the unlearned and retain-only models on the forget set) and model utility (MU, the harmonic mean of probability, ROUGE, and truth ratio on the retain, real-author, and world-fact sets); higher is better for both. On TOFU Forget05, CLUE reaches $\mathrm{FQ}/\mathrm{MU} = 0.92/0.88$ versus the strongest baseline SimNPO at $0.83/0.85$, and the gains persist on Forget10 ($0.87/0.84$ vs. $0.79/0.80$), where the MU gap widens. It stops within 5% of the oracle and transfers from TOFU to WMDP and MUSE.

Contributions. We (i) identify a three-phase structure (early, conflict, terminal) in LLM unlearning trajectories, observed across all seven evaluated methods with shared thresholds and with phase boundaries observable from ordinary backward-pass quantities (Section 4.3), and show that retain-loss damage accumulated over a long conflict phase cannot be undone by bounded controls within the unlearning horizon (Proposition B.2); (ii) design CLUE, a closed-loop controller whose dual-tower separation, conflict-driven gating, collapse-risk head, and anticipatory stopping each correspond to a phase or transition, with theoretical support from a single-step conflict-cost bound (Lemma B.1) and a stopping regret bound (Proposition B.3); (iii) evaluate on TOFU, WMDP, and MUSE with Llama-2/3 and Mistral models from 7B to 70B, with consistent FQ–MU gains, near-oracle stopping, and smaller degradation than fixed-schedule baselines under adversarial prompting, relearning, and INT4 quantization.

2 Related Work

Machine unlearning for LLMs. Machine unlearning [Cao and Yang, 2015, Bourtoule et al., 2021] has been extensively studied in the classification setting [Golatkar et al., 2020, Sekhari et al., 2021, Nguyen et al., 2025]. For LLMs, gradient ascent and its variants [Jang et al., 2023, Yao et al., 2024, Chen and Yang, 2023] reverse the gradient on the forget set but suffer from catastrophic collapse when applied aggressively [Zhang et al., 2024]. Negative Preference Optimization [Zhang et al., 2024] reframes unlearning through the lens of preference learning, treating the forget set as negative examples under a DPO-style objective, achieving exponentially slower divergence than gradient ascent. SimNPO [Fan et al., 2025] further removes the reference model dependency and achieves improved performance across difficulty levels. Representation-based methods such as RMU [Li et al., 2024] directly modify model representations to misdirect outputs on forget-set queries. Task arithmetic approaches [Ilharco et al., 2023] perform parameter-space subtraction to remove knowledge. Wang et al. [2025] analyze how unlearning objectives act through their gradients across instances, steps, and layers, and use this analysis to improve the objectives themselves. CLUE works one level up: given a forget/retain objective pair, it chooses the learning rate, retain weight, and

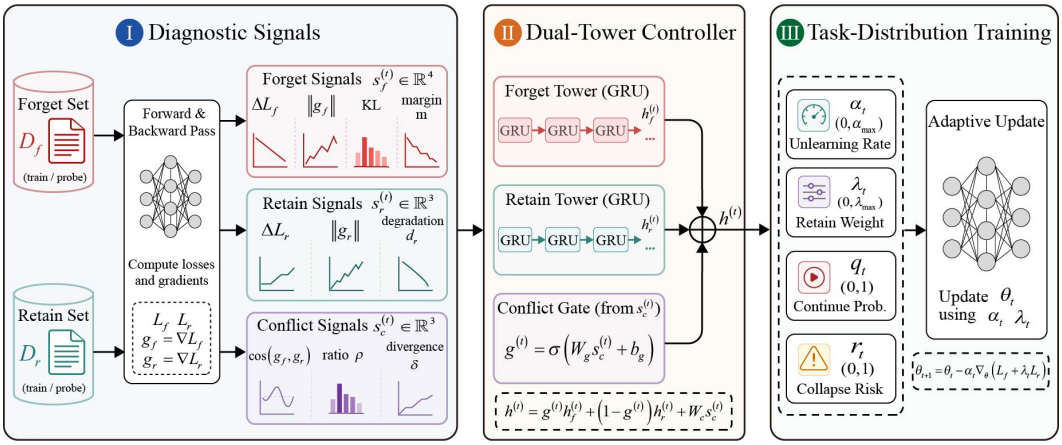

Figure 1: CLUE overview. Diagnostic signals from the forget set, retain set, and their gradient interaction feed a dual-tower controller with conflict gating and a collapse-risk head, which produces step-wise decisions on the unlearning rate, retain weight, and continuation, closing the loop around the model update.

stopping step online, and its conflict signal is a deliberately coarse per-step summary of the gradient interaction studied there. These methods are typically run with a fixed schedule that does not adapt to the trajectory. Two concurrent works also use the name CLUE, for circuit-based localization in LLM unlearning [Chen et al., 2026] and for retain-free class unlearning on mobile devices [Sayyed et al., 2026]; our method is a closed-loop trajectory controller and is unrelated to both.

Unlearning benchmarks and evaluation. TOFU [Maini et al., 2024] provides a controlled benchmark using fictitious author profiles, enabling precise measurement of forget quality and model utility. WMDP [Li et al., 2024] targets the removal of hazardous knowledge in biosecurity and cybersecurity domains. MUSE [Shi et al., 2025] evaluates unlearning of copyrighted and news content at scale. RWKU [Jin et al., 2024] benchmarks real-world knowledge unlearning with robustness evaluation. OpenUnlearning [Dorna et al., 2025] provides a unified evaluation framework across these benchmarks. Recent work has also highlighted the fragility of existing unlearning methods under adversarial attacks [Lynch et al., 2024, Yuan et al., 2025a, Zhou et al., 2025], relearning from small data subsets [Hu et al., 2025a], and quantization [Zhang et al., 2025]. Xu et al. [2026] use representation-level analysis to separate reversible from irreversible forgetting. Thaker et al. [2025] argue that current benchmarks overstate progress because forget and retain sets are nearly independent and results are sensitive to query phrasing, and BLUR [Hu et al., 2025b] introduces controlled forget-retain overlap to expose this gap. Our results are benchmark results in this sense and should not be read as certificates of deletion (Section 5).

Meta-learning for training adaptation. Meta-learning has been applied to a range of training adaptation problems, including learning rate scheduling [Xiong et al., 2022], learning curve extrapolation [Li and Zhao, 2026], sample weighting [Ren et al., 2018, Shu et al., 2019], curriculum design [Hacohen and Weinshall, 2019, Li et al., 2026a, Zhou et al., 2026], training-dynamics diagnosis [Li et al., 2025], and preference alignment [Li et al., 2026c,b]. These approaches largely target single-objective optimization where progress is monotone in a single quality dimension. Unlearning departs from this setting in two ways: the dual-objective structure produces gradient conflicts that are much weaker in alignment, and the risk of catastrophic representational collapse makes stopping a first-class control decision rather than a convergence criterion (Section 3.1).

3 Method

3.1 Unlearning as Dual-Objective Control

A dual-objective controller solves $\min_\theta \mathcal{L}_1(\theta) + \lambda \mathcal{L}_2(\theta)$, where $\mathcal{L}_1$ is the primary loss and $\mathcal{L}_2$ the constraint loss, along a trajectory by choosing a step size, a weight, and a stopping decision $(\alpha^{(t)}, \lambda^{(t)}, \text{stop}^{(t)})$ at each step from diagnostic signals. For unlearning, $\mathcal{L}_1 = \mathcal{L}_f$ and $\mathcal{L}_2 = \mathcal{L}_r$,

both minimized, with gradients $g_f = \nabla \mathcal{L}_f$ and $g_r = \nabla \mathcal{L}_r$. Existing controllers occupy low-conflict regimes: scheduling [Xiong et al., 2022] sets $\lambda \equiv 0$; reweighting [Ren et al., 2018, Shu et al., 2019] uses cooperative objectives ($\cos(g_1, g_2) \geq 0$); preference alignment [Li et al., 2026c] keeps conflict near zero. Unlearning differs along two axes: $\cos(g_f, g_r)$ is large and frequently negative, and collapse cannot be repaired within the unlearning horizon once the conflict phase is long enough (Proposition B.2). This positions unlearning at the high-conflict, low-recoverability corner of the design space and forces three architectural commitments: $\lambda^{(t)}$ must be conflict-aware, collapse must be anticipated rather than detected, and stopping must be a control action with its own supervision. Scheduling and reweighting controllers [Xiong et al., 2022, Ren et al., 2018, Shu et al., 2019] make none of these commitments; Appendix L shows that an alignment-style controller does not transfer to unlearning without them.

3.2 Problem Setup and Structural Analysis

The standard unlearning objective combines a forget loss $\mathcal{L}_f$ (ascent-style, e.g. GA, NPO) and a retain loss $\mathcal{L}_r$:

$$\min_{\theta} \; \mathcal{L}_f(\theta; D_f) + \lambda \, \mathcal{L}_r(\theta; D_r). \tag{1}$$

With $g_f = \nabla \mathcal{L}_f$ and $g_r = \nabla \mathcal{L}_r$, the update $\theta_{t+1} = \theta_t - \alpha(g_f + \lambda g_r)$ changes the retain loss at first order by $-\alpha[\lambda \|g_r\|^2 + \|g_f\| \|g_r\| \cos(g_f, g_r)]$, exposing two structural facts. *Single step*: if $\cos(g_f, g_r) \leq -c_0 < 0$ and $\lambda < \lambda^\star = c_0 \|g_f\| / \|g_r\|$, a sufficiently small step increases the retain loss, at a rate that grows with c_0 (Lemma B.1). This is a sufficient condition and we claim no converse, but it already implies that $\lambda^{(t)}$ should depend on $\cos(g_f^{(t)}, g_r^{(t)})$, which motivates conflict-driven gating. *Many steps*: a reactive rule that stops once retain loss has grown would suffice if later steps could undo the damage. Proposition B.2 gives a first-order, finite-horizon argument against this: damage accumulated over a conflict phase of length Δt grows linearly in Δt, whereas each corrective step with bounded (α', λ') removes at most a constant amount, so a long enough conflict phase cannot be repaired within the horizon $T_{\max}$. The proposition does not derive the empirical threshold $\delta^\star$ used in Section 4.3. It motivates anticipatory stopping and an explicit collapse-risk head supervised by lookahead labels.

3.3 Diagnostic Signals

We group the signals by what a dual-objective controller must track: progress on the forget objective, distance from the retained behavior, and the interaction between the two objectives. Table 1 lists all ten. Throughout, $\Delta x^{(t)} = x^{(t)} - x^{(t-1)}$ denotes a step-wise difference, and the probe batches $D_f^{\text{probe}}, D_r^{\text{probe}}$ contain 16 examples drawn at each step from the training portions of the forget and retain sets. All signals reuse the forward and backward passes that optimization already performs, and are normalized with running statistics from the first 10 steps (Appendix F).

Two signals need a formula. The logit margin

$$m^{(t)} = \mathbb{E}_{x \in D_f^{\text{probe}}} \Big[z_{(1)}(x; \theta^{(t)}) - z_{(2)}(x; \theta^{(t)}) \Big], \tag{2}$$

where $z_{(1)}, z_{(2)}$ are the largest and second-largest logits at the first answer position, first decreases as the model becomes uncertain on forgotten content and then collapses toward zero under over-unlearning. The mid-layer representation divergence

$$\delta^{(t)} = 1 - \text{CKA}\Big(H^{(t)}_{\lfloor L/2 \rfloor}(D_f^{\text{probe}}),\, H^{(t)}_{\lfloor L/2 \rfloor}(D_r^{\text{probe}}) \Big), \tag{3}$$

where $H_\ell^{(t)}$ denotes layer-ℓ activations mean-pooled over tokens and CKA is linear centered kernel alignment, rises abruptly at collapse onset. Rows are paired by batch index; since this pairing is arbitrary, we use $\delta^{(t)}$ as a trend signal rather than an absolute separation measure. Note that $\delta^{(t)}$ is a divergence while $\Delta\delta^{(t)} = \delta^{(t)} - \delta^{(t-1)}$ is its step-wise change. We keep the groups as an ordered tuple $\mathbf{d}^{(t)} = (\mathbf{s}_f^{(t)}, \mathbf{s}_r^{(t)}, \mathbf{s}_c^{(t)}) \in \mathbb{R}^{10}$, and the controller in Section 3.4 reads each group through a separate pathway. Neither $m^{(t)}$ nor $\delta^{(t)}$ is new as a quantity; what is specific to unlearning is that they become control inputs for anticipating collapse.

Table 1: The ten diagnostic signals. Δ is a step-wise difference; D_f, D_r indicate whether the forget or retain probe batch (16 examples) is used.

Signal	Definition	Data	Range	Role
Forget group $\mathbf{s}_f^{(t)} \in \mathbb{R}^4$ (progress)				
$\Delta\mathcal{L}_f^{(t)}$	$\mathcal{L}_f^{(t)} - \mathcal{L}_f^{(t-1)}$	D_f	$\mathbb{R}$	rate of forgetting
$\|g_f^{(t)}\|$	ℓ_2 norm of $\nabla\mathcal{L}_f$	D_f	≥ 0	forget pull
$D_{\mathrm{KL}}^{(t)}$	$D_{\mathrm{KL}}(\pi_{\theta^{(t)}} \,\|\, \pi_{\theta_0})$	D_f	≥ 0	output drift
$m^{(t)}$	Eq. (2)	D_f	≥ 0	collapse precursor
Retain group $\mathbf{s}_r^{(t)} \in \mathbb{R}^3$ (retention)				
$\Delta\mathcal{L}_r^{(t)}$	$\mathcal{L}_r^{(t)} - \mathcal{L}_r^{(t-1)}$	D_r	$\mathbb{R}$	rate of utility loss
$\|g_r^{(t)}\|$	ℓ_2 norm of $\nabla\mathcal{L}_r$	D_r	≥ 0	retain pull
$d_r^{(t)}$	$(\mathcal{L}_r^{(t)} - \mathcal{L}_r^{(0)})/\mathcal{L}_r^{(0)}$	D_r	≥ -1	cumulative utility loss
Conflict group $\mathbf{s}_c^{(t)} \in \mathbb{R}^3$ (interaction)				
$c^{(t)}$	$\cos(g_f^{(t)}, g_r^{(t)})$	both	$[-1, 1]$	gradient conflict
$\rho^{(t)}$	$\|g_f^{(t)}\|/\|g_r^{(t)}\|$	both	≥ 0	relative pull, enters $\lambda^\star$
$\delta^{(t)}$	Eq. (3)	both	$[0, 1]$	forget/retain divergence (trend)

3.4 Dual-Tower Controller with Conflict Gating

Rather than compressing $\mathbf{d}^{(t)}$ into a single recurrent state, we process forget and retain signals in two independent two-layer GRUs (hidden dimension 32 each) and let the conflict group govern their fusion:

$$\mathbf{h}_f^{(t)} = \mathrm{GRU}_f(\mathbf{s}_f^{(t)}, \mathbf{h}_f^{(t-1)}), \quad \mathbf{h}_r^{(t)} = \mathrm{GRU}_r(\mathbf{s}_r^{(t)}, \mathbf{h}_r^{(t-1)}). \tag{4}$$

The conflict group produces a gate, and is also skip-connected into the fused state so conflict magnitude is not diluted by the towers:

$$g^{(t)} = \sigma(\mathbf{w}_g^\top \mathbf{s}_c^{(t)} + b_g), \qquad \mathbf{h}^{(t)} = g^{(t)}\mathbf{h}_f^{(t)} + (1-g^{(t)})\mathbf{h}_r^{(t)} + \mathbf{W}_c\mathbf{s}_c^{(t)}. \tag{5}$$

Four linear heads with sigmoid activation produce the step-t decisions:

$$\alpha^{(t)} = \alpha_{\max}\sigma(\mathbf{w}_\alpha^\top \mathbf{h}^{(t)}),\ \lambda^{(t)} = \lambda_{\max}\sigma(\mathbf{w}_\lambda^\top \mathbf{h}^{(t)}),\ q^{(t)} = \sigma(\mathbf{w}_q^\top \mathbf{h}^{(t)}),\ r^{(t)} = \sigma(\mathbf{w}_r^\top \mathbf{h}^{(t)}). \tag{6}$$

The first three are standard (unlearning rate, retain weight, continuation probability). The fourth, $r^{(t)} \in [0, 1]$, is the collapse-risk head: it predicts whether the next update would enter the collapse regime, supervised during training by a lookahead label derived from $\Delta\mathcal{L}_r^{(t+1)}$ and $\Delta\delta^{(t+1)}$ (Eq. (9)). Stopping fires when either $q^{(t)} < \tau_q$ or $r^{(t)} > \tau_r$, letting the controller stop either because unlearning is complete or because further updates are unsafe. The model update then uses the step-t outputs:

$$\theta^{(t+1)} = \theta^{(t)} - \alpha^{(t)}\nabla_\theta\big[\mathcal{L}_f(\theta^{(t)}; D_f) + \lambda^{(t)}\mathcal{L}_r(\theta^{(t)}; D_r)\big]. \tag{7}$$

The full controller has $\sim$35K parameters, fewer than a single-tower GRU of hidden dimension 64 (47K); a parameter and wall-clock comparison with alternative cells and with a single-tower baseline equivalent to alignment-style controllers is in Appendix K.

3.5 Task-Distribution Training

We train the controller over a distribution of unlearning tasks, evaluating each decision on question-answer pairs that the inner loop never touches. The within-task train/validation split follows standard support/query separation in meta-learning [Finn et al., 2017, Ren et al., 2018]; task diversity comes from varying forget-set content, forget-set size (1%–20%), and knowledge type (factual, stylistic, entity).

At each inner step, a virtual update is computed,

$$\theta_t'(\phi) = \theta_t - \alpha^{(t)}\nabla_{\theta_t}\big[\mathcal{L}_f(\theta_t; D_f^{\mathrm{tr}}) + \lambda^{(t)}\mathcal{L}_r(\theta_t; D_r^{\mathrm{tr}})\big], \tag{8}$$

and the controller is trained to minimize a meta-objective evaluated on the validation split:

$$\mathcal{L}_{\text{meta}}(\phi) = \mathcal{L}_f^{\text{val}}(\theta_t'(\phi)) + \gamma\, \mathcal{L}_r^{\text{val}}(\theta_t'(\phi)) + \xi\, \text{BCE}(r^{(t)}, y_c^{(t)}) + \zeta\, \text{BCE}(q^{(t)}, y_q^{(t)}), \tag{9}$$

with $y_c^{(t)} = \mathbb{I}[\Delta\mathcal{L}_r^{(t+1)} > \varepsilon_r \vee \Delta\delta^{(t+1)} > \varepsilon_\delta]$ labeling the step as collapse-positive if retain loss or representation divergence rises sharply at the next step. The third term supervises the collapse-risk head with a one-step lookahead, converting stopping from a reactive criterion into a predictive one. The fourth term supervises the continuation head with $y_q^{(t)} = \mathbb{I}[t < t^\star]$, where $t^\star$ is the step of the training trajectory that maximizes $\text{FQ}\cdot\text{MU}$. We use $\zeta = 0.5$, selected on meta-validation, and no class weighting for either BCE term; sensitivity to γ, ξ, ζ, τ_r, and the remaining meta-hyperparameters is reported in Appendix J (Table 18). Training uses first-order gradients through Eq. (8) [Finn et al., 2017, Ren et al., 2018]; the actual model update then uses the refreshed controller outputs. The full procedure is Algorithm 1.

Evaluating on held-out question-answer pairs removes the incentive to inflate the forget loss on the training samples themselves, a source of superficial forgetting [Lynch et al., 2024].

At inference, no hyperparameter search is needed: the trained controller is applied directly with the stopping rule above, adding about 2.4% per-step wall-clock time relative to retain-aware baselines on Llama-2-7B (Appendix E).

4 Experiments

4.1 Setup

Benchmarks, models, baselines. We evaluate on TOFU [Maini et al., 2024] (Forget01/05/10), WMDP [Li et al., 2024], and MUSE [Shi et al., 2025], with Llama-2-7B [Touvron et al., 2023], Llama-3-8B [Grattafiori et al., 2024], Mistral-7B-v0.3 [Jiang et al., 2023] as primary models, plus Llama-2-13B/70B (LoRA, rank 16) for scalability. Baselines: GA [Yao et al., 2024], GradDiff [Yao et al., 2024], KL [Maini et al., 2024], NPO [Zhang et al., 2024], SimNPO [Fan et al., 2025], RMU [Li et al., 2024], each grid-tuned per benchmark (search ranges in Appendix F). The 7B and 8B models are unlearned with full-parameter updates; 13B and 70B use LoRA.

Metrics and CLUE training. TOFU: FQ and MU (both higher better). WMDP: accuracy (lower, random $= 25\%$) and retained MMLU (higher). MUSE: $|\text{PrivLeak}|$ (closer to zero is better) and KnowMem (higher). Numbers are means over 5 seeds, matched across methods. The controller is trained on Llama-2-7B over the TOFU task distribution with 500 meta-updates, one every $K = 10$ inner steps or at the end of a rollout ($T_{\max} = 100$, $\alpha_{\max} = 5\times10^{-5}$, $\lambda_{\max} = 10$, $\gamma = 1.0$, $\xi = \zeta = 0.5$, $\tau_q = 0.5$, $\tau_r = 0.8$, Adam at $\eta = 10^{-3}$), about 8 hours on one A40. Of the 200 TOFU profiles, 160 are used for meta-training and 40 for meta-validation. Standard TOFU evaluation then uses all 200 profiles, so our TOFU numbers are held out at the level of question-answer pairs (a 14/6 split per profile) but not at the level of profiles. Hyperparameter ranges and baseline configurations are in Appendix F, and a per-step cost breakdown in Appendix E.

4.2 Main Results

TOFU. Table 2 shows consistent gains on all three forget-set sizes. At Forget05, CLUE reaches $\text{FQ}/\text{MU} = 0.92/0.88$ versus the best baseline SimNPO at $0.83/0.85$. The gains persist on Forget10 ($0.87/0.84$ vs. $0.79/0.80$), and the MU gap widens, consistent with our structural analysis: larger forget sets produce more frequent conflict and higher collapse risk, the two regimes in which fixed-strategy baselines fail and adaptive control matters most. Because seeds are matched, we also compare CLUE and SimNPO seed by seed: CLUE is ahead on all five seeds for both metrics at both Forget05 and Forget10, with mean paired gains of $+0.09/+0.03$ (Forget05) and $+0.08/+0.04$ (Forget10) in FQ/MU (Appendix H.4). Five seeds do not support a formal significance test, and we do not claim one. Giving SimNPO 60 additional cosine-scheduled configurations raises it to $0.85/0.86$ on Forget05, which narrows but does not close the gap (Appendix H.5). Per-model breakdowns are in Appendix H, and Pareto frontiers in Figure 4(a).

WMDP and MUSE. Table 3 reports results on the remaining benchmarks. On WMDP, CLUE reduces hazardous-knowledge accuracy to 28.3% (random baseline 25%) while preserving 95.1%

Table 2: TOFU benchmark. FQ and MU higher is better; best in bold. Mean±std over 5 seeds.

Method	Forget01		Forget05		Forget10	
	FQ ↑	MU ↑	FQ ↑	MU ↑	FQ ↑	MU ↑
Retrained	1.00	1.00	1.00	1.00	1.00	1.00
GA	0.78±0.02	0.91±0.01	0.72±0.03	0.74±0.04	0.68±0.04	0.61±0.05
GradDiff	0.81±0.02	0.92±0.01	0.76±0.02	0.80±0.03	0.71±0.03	0.68±0.04
KL	0.80±0.02	0.93±0.01	0.74±0.02	0.82±0.02	0.70±0.03	0.72±0.03
NPO	0.85±0.02	0.93±0.01	0.80±0.02	0.84±0.02	0.76±0.02	0.78±0.02
SimNPO	0.87±0.01	0.94±0.01	0.83±0.02	0.85±0.02	0.79±0.02	0.80±0.02
RMU	0.83±0.02	0.94±0.01	0.78±0.02	0.87±0.02	0.74±0.02	0.82±0.02
CLUE	**0.91±0.02**	**0.95±0.01**	**0.92±0.02**	**0.88±0.02**	**0.87±0.03**	**0.84±0.02**

Table 3: WMDP-Bio: multiple-choice accuracy (lower better, random = 25%); MMLU as % of pretrained score (higher better, original = 100%). MUSE: |PrivLeak| (closer to 0 is better) and KnowMem (higher). Mean over 5 seeds. CLUE uses a controller meta-trained on each target benchmark.

Method	WMDP-Bio		MUSE-Books		MUSE-News	
	Acc ↓	MMLU (% of orig.) ↑	|PrivLeak| ↓	KnowMem ↑	|PrivLeak| ↓	KnowMem ↑
Original	64.8	100.0	0.89	1.00	0.85	1.00
GA	32.6	88.5	0.12	0.78	0.15	0.74
GradDiff	31.4	90.2	0.10	0.82	0.13	0.79
NPO	30.8	91.5	0.08	0.85	0.10	0.83
SimNPO	29.5	92.1	0.07	0.87	0.09	0.85
RMU	30.1	92.8	0.09	0.86	0.11	0.84
CLUE	**28.3**	**95.1**	**0.04**	**0.91**	**0.06**	**0.89**

of the original MMLU score, a stronger FQ/MU trade-off than any baseline. On MUSE-Books and MUSE-News, CLUE achieves the lowest privacy leakage (0.04 and 0.06) and the highest knowledge retention (0.91 and 0.89), confirming that the adaptive control remains effective when the forget-retain boundary is less discrete than TOFU.

4.3 The Three-Phase Structure of LLM Unlearning

We test the three-phase claim on the six baselines, three forget sizes, and task arithmetic as a structurally different editing method. Let $\bar{c}^{(t)}, \bar{\delta}^{(t)}, \bar{m}^{(t)}$ be 5-step running averages of the cosine, the representation divergence, and the logit margin. The early-to-conflict transition t_1 is the first step with $\bar{c}^{(t)} < -0.1$ and $\Delta\bar{\delta}^{(t)} > 0$ for three consecutive steps; the conflict-to-terminal transition t_2 is the first step with $\bar{m}^{(t)} < 0.5\,\bar{m}^{(0)}$ and $\bar{\delta}^{(t)} > \delta^\star$. The threshold $\delta^\star$ is calibrated on trajectories from the 40 meta-validation profiles, is used only for this analysis, and is never an input to the controller; Proposition B.2 motivates the existence of such a threshold but does not derive its value. The thresholds $(0.1, 0.5, \delta^\star)$ are shared across all methods.

Figure 2 and Table 4(a) show the structure. A recurring three-phase pattern appears across all seven methods when run to $T_{\max}$, except that about 6% of Forget10 trajectories lack a distinct early phase (Appendix N); the early phase ends with $c^{(t)}$ turning sharply negative, and retain loss never recovers after the terminal phase begins. NPO and SimNPO delay the conflict phase relative to GA but do not avoid it, and RMU shifts where the threshold is crossed. Task arithmetic, applied as $\theta_0 - \eta\,\tau_f$ with the forget task vector τ_f and η swept over 100 coefficient values, crosses $\delta^\star$ on all five seeds with the original thresholds and no retuning. The differences across methods are quantitative rather than structural. We have not tested MoE, vision-language, or sub-1B models.

The preferred step-wise action also differs across phases. In the early phase both $c^{(t)}$ and $\delta^{(t)}$ are near zero, and the oracle action keeps $\alpha^{(t)}$ at its upper bound with $\lambda^{(t)}$ near zero. In the conflict phase, Lemma B.1 shows that keeping $\lambda^{(t)}$ below $\lambda^\star = c_0\|g_f\|/\|g_r\|$ lets small steps increase the

Table 4: (a) Phase boundaries on TOFU Forget05 (Llama-2-7B); $\int \Delta\mathcal{L}_r$ is cumulative retain-loss growth in the conflict phase (not computed for task arithmetic, whose "steps" are coefficient values). (b) Stopping step of the oracle (step maximizing FQ · MU, computed after the run) and of CLUE.

(a)

Method	Early	Conflict	$\int \Delta\mathcal{L}_r$
GA	1–8	9–31	0.42
GradDiff	1–11	12–38	0.31
KL	1–13	14–42	0.28
NPO	1–15	16–48	0.21
SimNPO	1–16	17–52	0.18
RMU	1–14	15–45	0.23
Task arithmetic	1–4	5–22	–

(b)

Setting	Oracle	CLUE	Gap
Forget01	38	39	+2.6%
Forget05	50	52	+4.0%
Forget10	62	65	+4.8%

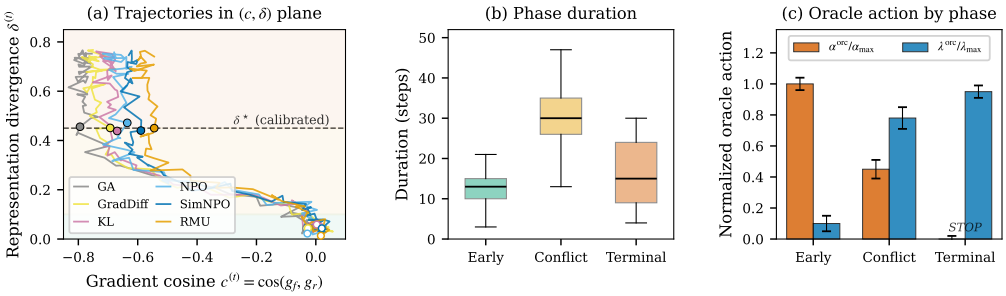

Figure 2: Three-phase structure of LLM unlearning. Left: trajectories of six baselines on TOFU Forget05 in the $(c^{(t)}, \delta^{(t)})$ plane; dashed line marks the empirical threshold $\delta^\star$, calibrated on meta-validation profiles. Middle: phase-duration distribution over 200 trajectories. Right: oracle-search action per phase, normalized by its upper bound; the qualitative change across phases motivates closed-loop control.

retain loss. In the terminal phase, retain loss no longer recovers, so the preferred action is to stop. A fixed-schedule method commits to a single (α, λ) and a fixed step count, so it can match at most one phase's policy.

4.4 Adaptive Control Behavior

We now verify that CLUE's learned policy realizes the phase-dependent control prescribed in Section 4.3. Figure 3 shows the controller's response to diagnostic signals in real time on a representative TOFU Forget05 trajectory. The top row visualizes the stopping mechanism: FQ and MU curves alongside the oracle optimum (star) and the step CLUE selects (circle); across the three forget sizes, CLUE stops within 5% of the oracle (Table 4(b)). The oracle step is computed only after a trajectory is complete and is never available to the controller. The bottom row tracks signals and controller outputs along a single run.

The three observed regimes match the phase structure of Section 4.3 step-for-step. In the early phase (steps 1–15), $\cos(g_f, g_r)$ is near zero and $\delta^{(t)}$ is flat; the controller keeps $\alpha^{(t)}$ at its upper bound and $\lambda^{(t)}$ low, which matches the early-phase oracle action. In the conflict phase (from step 15), $\cos(g_f, g_r)$ turns sharply negative and $\delta^{(t)}$ rises; the controller reduces $\alpha^{(t)}$ and raises $\lambda^{(t)}$, as Lemma B.1 suggests. At step 52 the collapse-risk head $r^{(t)}$ crosses τ_r while $q^{(t)}$ is still above τ_q, one step before $\delta^{(t)}$ would cross $\delta^\star$, so unlearning stops before the terminal phase. The controller is never told the phase structure; its outputs converge to it under the meta-objective in Eq. (9).

4.5 Diagnostic Signal Informativeness

The regret bound in Proposition B.3 grows with σ, the residual variance after predicting the FQ/MU utility from the diagnostic state $\mathbf{d}^{(t)}$. To empirically anchor this quantity, we train linear probes to predict FQ and MU from $\mathbf{d}^{(t)}$ alone, using 200 unlearning runs on TOFU and evaluating on 50

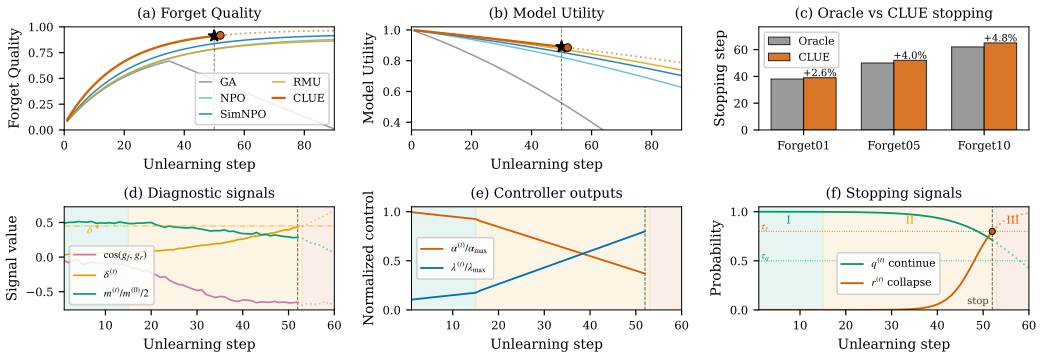

Figure 3: Controller behavior on TOFU Forget05 (Llama-2-7B). Top: FQ, MU, and oracle-vs-CLUE stopping across baselines. Bottom: diagnostic signals, controller outputs, and stopping signals $q^{(t)}, r^{(t)}$ along one trajectory. Phases are shaded green/orange/red; the dashed line marks the stop, and dotted lines show the run continued without stopping.

Table 5: Linear probing R^2 for predicting FQ and MU from diagnostic signals on held-out runs. Combining groups yields $R^2 \geq 0.88$.

Signal group	R^2 for FQ	R^2 for MU
Forget signals only ($\mathbf{s}_f$)	0.82	0.31
Retain signals only ($\mathbf{s}_r$)	0.24	0.79
Conflict signals only ($\mathbf{s}_c$)	0.41	0.52
All signals ($\mathbf{d}$)	**0.91**	**0.88**

held-out runs. Table 5 reports R^2 per signal group. Forget and retain groups are complementary: forget signals carry most of the information about FQ (0.82) but little about MU (0.31), while retain signals are the opposite. Conflict signals alone are moderately predictive of both. When all groups are combined, R^2 is at least 0.88 for both targets, which suggests that the residual variance of a linear predictor is at most about 12% of the variance of each metric and that the first term of the regret bound is small.

4.6 Generalization, Scalability, and Ablations

Cross-scenario transfer. The controller is trained on TOFU only. When applied zero-shot to WMDP-Bio and MUSE, it reaches $29.1/94.5$ on WMDP-Bio and $|\text{PrivLeak}|$/KnowMem of $0.05/0.89$ (MUSE-Books) and $0.07/0.87$ (MUSE-News), within 0.8 points (WMDP) and 0.02 (MUSE) of the target-specific controllers in Table 3, and substantially above every fixed-strategy baseline in Table 3. Full numbers in Appendix H.6.

Scalability. Figure 4(b) reports TOFU Forget05 across five models from 7B to 70B. The controller trained on Llama-2-7B is applied with unchanged weights: FQ/MU stay in the 0.88–0.93 / 0.86–0.89 range, with full-parameter updates for Llama-2-7B, Llama-3-8B, and Mistral-7B and LoRA (rank 16) for Llama-2-13B and 70B; for 70B only the rate bound $\alpha_{\max}$ is reduced. Total per-step overhead (signal extraction plus controller) is 2.4%, 1.9%, and 0.4% for 7B, 13B, and 70B; controller inference alone is about 0.02% of step time because its input is 10-dimensional at every scale. We have not tested full-parameter unlearning above 8B, transfer from larger to smaller models, or MoE and sub-1B models. Full numbers are in Appendix H.7.

Ablation. Figure 4(c) shows four patterns on Forget05. First, forget signals matter most for FQ, retain signals for MU, and conflict signals for both. Second, the two unlearning-specific signals introduced in Section 3.3 each account for ≥ 0.02 of performance in at least one metric, and replacing them with the membership-inference proxy used in alignment-style controllers recovers only part of the gap. Third, fixing either $\alpha^{(t)}$ or $\lambda^{(t)}$ degrades MU measurably, and removing the collapse-risk head forces stopping to rely on $q^{(t)}$ alone, causing late-phase over-unlearning. Fourth, the structural

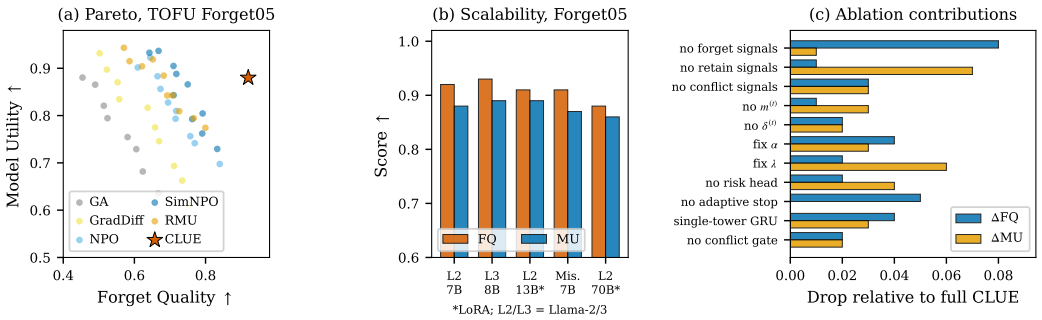

Figure 4: Key results. (a) Pareto frontier on TOFU Forget05: baselines (dots) trace frontiers via hyperparameter sweeps; CLUE (star) reaches a non-dominated point without per-task tuning. (b) Scalability: same TOFU-7B controller applied across five model scales. (c) Ablation: drop in FQ (blue) and MU (orange) when each component is removed.

choices (dual tower, conflict gating, skip connection) together add 0.04 in FQ and 0.03 in MU over a single-tower GRU that reads the same ten signals with more parameters (47K vs. 35K); the single-tower variant is the direct analogue of alignment-style controllers. A rule-based controller with hand-set thresholds on the same signals reaches $0.87/0.83$ (Appendix M). Full ablation numbers for 15 configurations are in Appendix H.8, and one-at-a-time sensitivity of the meta-hyperparameters, plotted against the baseline frontier, is in Appendix J.

Robustness. Across three robustness probes, CLUE shows smaller degradation than fixed-schedule baselines. Adversarial prompting extracts 6.8% of forget content (vs. SimNPO 11.7%), forget accuracy after relearning on 10% of the forget data for 10 AdamW steps is 33.4% (vs. 42.8%), and FQ drops by 0.03 under INT4 quantization (vs. 0.05–0.12 for baselines). A residual gap to the retrained-from-scratch reference (2.1% / 18.5% / 1.00) remains; we do not claim CLUE matches that reference, only that it consistently narrows the gap. We attribute this to the controller stopping before the terminal phase, not to a different mechanism of forgetting. These probes test recoverability under our protocol; they are not a head-to-head replication of the recovery analysis of Xu et al. [2026], and they do not address the forget-retain independence issue raised by Thaker et al. [2025]. Full numbers are in Appendices H.9 and I.

5 Conclusion

Across the seven unlearning methods we evaluate, LLM unlearning trajectories pass through the same three phases, and the phase boundaries can be read from quantities the optimizer already computes. CLUE aligns control with this structure through a dual-tower controller fused by a conflict-driven gate, with an explicit collapse-risk head that fires before the empirical threshold is crossed. Empirically, it improves the forget–utility trade-off over grid-tuned fixed schedules, stops near the oracle, and transfers across benchmarks and model scales without retraining.

Limitations. *Scope.* The phase structure is observed on Llama-2/3 and Mistral (7B to 70B) for six gradient-based methods and task arithmetic; MoE, vision-language, sub-1B models, and other editing mechanisms are untested, $\delta^\star$ will likely need recalibration there, about 6% of Forget10 trajectories lack a separable early phase, and full-parameter evidence stops at 8B. *Evaluation.* TOFU results are held out at the level of question-answer pairs but not of profiles, $\delta^\star$ is calibrated on the meta-validation profiles, and $\delta^{(t)}$ may depend on batch composition and row pairing; a strict profile-level split remains to be done. Our FQ/MU numbers are benchmark performance in the sense of Thaker et al. [2025], not certificates of deletion, and we have not evaluated under controlled forget-retain overlap [Hu et al., 2025b]. *Compute.* Meta-training costs about 8 A40-hours once, versus about 2.2 hours for a 25-run grid search per baseline and task (Appendix E), so reuse breaks even after about four tasks, while a fixed baseline is cheaper for a single task; lacking standardized GPU-hour logs, we make no equal-compute claim. *Theory.* Our results are first-order with instance-dependent constants (Appendix B), so we treat them as structural. *Safety.* On WMDP, a stopping decision is not a safety certificate; deployment requires independent extraction and relearning audits. Failure modes are discussed in Appendix N.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China under Grant 62606363. The authors declare no competing interests.

References

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *IEEE symposium on security and privacy*, pages 141–159. IEEE, 2021.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *IEEE symposium on security and privacy*, pages 463–480. IEEE, 2015.

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 *USENIX Security 21*, pages 2633–2650, 2021.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. In *ICLR*, 2023.

Hang Chen, Jiaying Zhu, Xinyu Yang, and Wenya Wang. CLUE: Conflict-guided localization for LLM unlearning framework. In *ICLR*, pages 52000–52021, 2026.

Jiaao Chen and Diyi Yang. Unlearn what you want to forget: Efficient unlearning for llms. In *EMNLP*, pages 12041–12052, 2023.

Vineeth Dorna, Anmol Mekala, Wenlong Zhao, Andrew McCallum, Zico Kolter, Zachary Lipton, and Pratyush Maini. OpenUnlearning: Accelerating LLM unlearning via unified benchmarking of methods and metrics. In *NeurIPS*, 2025.

Ronen Eldan and Mark Russinovich. Who's Harry Potter? approximate unlearning in LLMs. *arXiv preprint arXiv:2310.02238*, 2023.

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 *NeurIPS*, pages 1540–1567, 2025.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135. PMLR, 2017.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *CVPR*, pages 9304–9312, 2020.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. In *ICML*, pages 2535–2544, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022.

Shengyuan Hu, Yiwei Fu, Steven Wu, and Virginia Smith. Unlearning or obfuscating? jogging the memory of unlearned llms via benign relearning. In *International Conference on Learning Representations*, volume 2025, pages 8857–8888, 2025a.

Shengyuan Hu, Neil Kale, Pratiksha Thaker, Yiwei Fu, Steven Wu, and Virginia Smith. BLUR: A benchmark for LLM unlearning robust to forget-retain overlap. *arXiv preprint arXiv:2506.15699*, 2025b.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *ICLR*, 2023.

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 *ACL*, pages 14389–14408, 2023.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

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 *NeurIPS*, volume 37, pages 98213–98263, 2024.

Mengyang Li and Pinlong Zhao. Difficulty-aware learning curve extrapolation. In *AAAI*, page 23021–23029, 2026.

Mengyang Li, Xiaoling Zhou, and Ou Wu. Delving into the training dynamics for image classification. *IEEE Transactions on Image Processing*, pages 6783–6798, 2025.

Mengyang Li, Haozhan Geng, Zhong Zhang, and Shuang Liu. Dual-difficulty curriculum learning for direct preference optimization. In *KDD*, page 2662–2673, 2026a.

Mengyang Li, Shuang Liu, and Zhong Zhang. Layer-wise gradient disentanglement: Decoupling semantics and preferences in direct preference optimization. In *ICML*, 2026b.

Mengyang Li, Pinlong Zhao, and Zhong Zhang. Aligner, diagnose thyself: A meta-learning paradigm for fusing intrinsic feedback in preference alignment. In *ICLR*, 2026c.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. In *ICML*, volume 235, pages 28525–28550. ML Research Press, 2024.

Aengus Lynch, Phillip Guo, Aidan Ewart, Stephen Casper, and Dylan Hadfield-Menell. Eight methods to evaluate robust unlearning in llms. *arXiv preprint arXiv:2402.16835*, 2024.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary Chase Lipton, and J Zico Kolter. Tofu: A task of fictitious unlearning for llms. In *COLM*, 2024.

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

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, pages 4334–4343. PMLR, 2018.

A. Q. M. Sazzad Sayyed, Nathaniel D. Bastian, Michael De Lucia, Ananthram Swami, and Francesco Restuccia. CLUE: Bringing machine unlearning to mobile devices. In *WACV*, pages 3750–3759, 2026.

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. In *NeurIPS*, volume 34, pages 18075–18086, 2021.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah Smith, and Chiyuan Zhang. Muse: Machine unlearning six-way evaluation for language models. In *ICLR*, pages 27797–27818, 2025.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *NeurIPS*, volume 32, 2019.

Pratiksha Thaker, Shengyuan Hu, Neil Kale, Yash Maurya, Zhiwei Steven Wu, and Virginia Smith. Position: LLM unlearning benchmarks are weak measures of progress. In *IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 520–533, 2025.

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

Qizhou Wang, Jin Zhou, Zhanke Zhou, Saebyeol Shin, Bo Han, and Kilian Weinberger. Rethinking LLM unlearning objectives: A gradient perspective and go beyond. In *ICLR*, pages 61897–61931, 2025.

Yuanhao Xiong, Li-Cheng Lan, Xiangning Chen, Ruochen Wang, and Cho-Jui Hsieh. Learning to schedule learning rate with graph neural networks. In *ICLR*, 2022.

Xiaoyu Xu, Xiang Yue, Yang Liu, Qingqing Ye, Huadi Zheng, Peizhao Hu, Minxin Du, and Haibo Hu. Unlearning isn't deletion: Investigating reversibility of machine unlearning in LLMs. In *ICML*, pages 142657–142704, 2026.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. In *NeurIPS*, volume 37, pages 105425–105475, 2024.

Hongbang Yuan, Zhuoran Jin, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. Towards robust knowledge unlearning: An adversarial framework for assessing and improving unlearning robustness in large language models. In *AAAI*, pages 25769–25777, 2025a.

Xiaojian Yuan, Tianyu Pang, Chao Du, Kejiang Chen, Weiming Zhang, and Min Lin. A closer look at machine unlearning for large language models. In *ICLR*, pages 49483–49508, 2025b.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. In *COLM*, 2024.

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 *ICLR*, pages 74925–74948, 2025.

Xiaoling Zhou, Mingjie Zhang, Zhemg Lee, Yuncheng Hua, Chengli Xing, Wei Ye, Flora Salim, and Shikun Zhang. Boosting resilience of large language models through causality-driven robust optimization. In *NeurIPS*, pages 55821–55845, 2025.

Xiaoling Zhou, Shuaiyu Zhou, Zhemg Lee, Tao Chen, Xirui Li, Peng Chen, Jie Jiang, Wei Ye, and Shikun Zhang. Tailoring the training: Difficulty-aware learning strategy allocation for large language models. In *ICML*, 2026.

A Algorithm

Algorithm 1 gives the training procedure; Algorithm 2 gives inference. Both call the dual-tower controller ψ_ϕ which encapsulates Eqs. (4)–(6): separate GRU updates, conflict gate, fusion, and the four heads.

Algorithm 1 CLUE Training

Require: Task distribution $p(\mathcal{T})$, pretrained θ_0, hyperparameters $\{\eta, \gamma, \xi, \zeta, \tau_q, \tau_r, K\}$
Initialize controller ϕ; $\mathcal{L}_{\text{meta}} \leftarrow 0$
while fewer than 500 meta-updates **do**
  Sample $\mathcal{T}_i \sim p(\mathcal{T})$; split $\{D_f^{\text{tr}}, D_f^{\text{val}}, D_r^{\text{tr}}, D_r^{\text{val}}\}$; reset $\theta, \mathbf{h}_f, \mathbf{h}_r$
  for $t = 1, \ldots, T_{\max}$ **do**
    $(\alpha^{(t)}, \lambda^{(t)}, q^{(t)}, r^{(t)}) \leftarrow \psi_\phi(\mathbf{d}^{(t)})$ with $\mathbf{d}^{(t)}$ from θ on $(D_f^{\text{tr}}, D_r^{\text{tr}})$
    if $q^{(t)} < \tau_q$ or $r^{(t)} > \tau_r$ **then** update ϕ with any accumulated $\mathcal{L}_{\text{meta}}$ and **break**
    $\theta'(\phi) \leftarrow \theta - \alpha^{(t)} \nabla[\mathcal{L}_f + \lambda^{(t)} \mathcal{L}_r]$ on D^{tr}; form labels $y_c^{(t)}, y_q^{(t)}$
    $\mathcal{L}_{\text{meta}} \leftarrow \mathcal{L}_{\text{meta}} + \mathcal{L}_f^{\text{val}} + \gamma \mathcal{L}_r^{\text{val}} + \xi\, \text{BCE}(r^{(t)}, y_c^{(t)}) + \zeta\, \text{BCE}(q^{(t)}, y_q^{(t)})$ at $\theta'(\phi)$
    if $t \bmod K = 0$ or $t = T_{\max}$: $\phi \leftarrow \phi - \eta \nabla_\phi \mathcal{L}_{\text{meta}}$ (first-order); $\mathcal{L}_{\text{meta}} \leftarrow 0$
    Commit actual update $\theta \leftarrow \theta'(\phi)$ with refreshed controller outputs
  end for
end while
return ϕ

Algorithm 2 CLUE Inference

Require: Trained ψ_ϕ, model θ_0, sets D_f, D_r, thresholds τ_q, τ_r
$\theta \leftarrow \theta_0$; reset hidden states
for $t = 1, \ldots, T_{\max}$ **do**
  $(\alpha^{(t)}, \lambda^{(t)}, q^{(t)}, r^{(t)}) \leftarrow \psi_\phi(\mathbf{d}^{(t)})$
  if $q^{(t)} < \tau_q$ or $r^{(t)} > \tau_r$ **then break**
  $\theta \leftarrow \theta - \alpha^{(t)} \nabla[\mathcal{L}_f(D_f) + \lambda^{(t)} \mathcal{L}_r(D_r)]$
end for
return θ

B Theoretical Analysis

This appendix provides three results that map onto the three architectural commitments of CLUE. Lemma B.1 bounds single-step retain-loss growth under gradient conflict, motivating the conflict-driven gate. Proposition B.2 gives a first-order, finite-horizon recovery bound showing why damage accumulated over a long conflict phase cannot be undone within the unlearning horizon, motivating the explicit collapse-risk head and anticipatory stopping. Proposition B.3 bounds the controller's stopping regret in terms of the informativeness of the diagnostic signals, giving a formal anchor for the empirical linear-probing analysis of Appendix D.

Throughout this appendix we adopt the following notation. At step t, forget and retain gradients are $g_f^{(t)} = \nabla \mathcal{L}_f(\theta^{(t)})$ and $g_r^{(t)} = \nabla \mathcal{L}_r(\theta^{(t)})$. The cosine between them is $c^{(t)} = \cos(g_f^{(t)}, g_r^{(t)})$. The step is

$$\theta^{(t+1)} = \theta^{(t)} - \alpha^{(t)} \big(g_f^{(t)} + \lambda^{(t)} g_r^{(t)}\big), \tag{10}$$

where $\alpha^{(t)} \in (0, \alpha_{\max}]$ and $\lambda^{(t)} \in [0, \lambda_{\max}]$ are controller outputs. We drop the superscript (t) where clear from context.

B.1 Single-Step Conflict Cost

We first record the one-step bound that motivates the conflict-driven gate. The result is a direct consequence of smoothness and Cauchy-Schwarz; we present it as a lemma that anchors the multi-step analysis to follow rather than as an independent contribution.

Lemma B.1 (Single-step conflict cost)**.** *Suppose $\mathcal{L}_r$ is μ-smooth, that is, $\mathcal{L}_r(\theta + v) \leq \mathcal{L}_r(\theta) + \langle \nabla \mathcal{L}_r(\theta), v \rangle + \frac{\mu}{2}\|v\|^2$ for all θ, v. Fix step t and assume $c^{(t)} \leq -c_0 < 0$ and $\|g_r^{(t)}\| \geq m > 0$. Define the conflict threshold weight*

$$\lambda^\star \;=\; c_0 \,\frac{\|g_f^{(t)}\|}{\|g_r^{(t)}\|}. \tag{11}$$

Whenever $\lambda^{(t)} < \lambda^\star$, the one-step change in the retain loss under the update Eq. (10) *is bounded below by*

$$\mathcal{L}_r(\theta^{(t+1)}) - \mathcal{L}_r(\theta^{(t)}) \;\geq\; \alpha^{(t)}\, c_0\, \|g_f^{(t)}\|\, \|g_r^{(t)}\| \left(1 - \lambda^{(t)}/\lambda^\star\right) \;-\; \tfrac{\mu}{2}\,(\alpha^{(t)})^2 \left\|g_f^{(t)} + \lambda^{(t)} g_r^{(t)}\right\|^2. \tag{12}$$

In particular, for $\alpha^{(t)} < 2c_0\|g_f^{(t)}\|\|g_r^{(t)}\|(1 - \lambda^{(t)}/\lambda^\star)/(\mu\|g_f^{(t)} + \lambda^{(t)} g_r^{(t)}\|^2)$ the retain loss strictly increases. The condition $\lambda^{(t)} < \lambda^\star$ is sufficient, not necessary; no converse is claimed.

Proof. Let $v = -\alpha^{(t)}(g_f^{(t)} + \lambda^{(t)} g_r^{(t)})$ denote the step. By μ-smoothness of $\mathcal{L}_r$,

$$\mathcal{L}_r(\theta^{(t+1)}) - \mathcal{L}_r(\theta^{(t)}) \;\geq\; \langle g_r^{(t)}, v \rangle - \tfrac{\mu}{2}\|v\|^2, \tag{13}$$

which follows from the two-sided form of smoothness $|\mathcal{L}_r(\theta + v) - \mathcal{L}_r(\theta) - \langle g_r, v\rangle| \leq \frac{\mu}{2}\|v\|^2$. The linear term evaluates to

$$\langle g_r^{(t)}, v \rangle = -\alpha^{(t)} \big[\, c^{(t)} \|g_f^{(t)}\|\, \|g_r^{(t)}\| + \lambda^{(t)} \|g_r^{(t)}\|^2 \,\big].$$

Using $c^{(t)} \leq -c_0$ and factoring $\|g_r^{(t)}\|$,

$$\langle g_r^{(t)}, v \rangle \geq \alpha^{(t)}\, c_0\, \|g_f^{(t)}\|\, \|g_r^{(t)}\| \left(1 - \lambda^{(t)}/\lambda^\star\right),$$

where the last line uses $\lambda^{(t)}\|g_r^{(t)}\|^2 = c_0\|g_f^{(t)}\|\|g_r^{(t)}\| \cdot \lambda^{(t)}/\lambda^\star$, which follows from Eq. (11). Substituting into Eq. (13) gives Eq. (12). □

The lemma shows that whenever $c^{(t)} \leq -c_0$, a retain weight below $\lambda^\star$ is enough for a small step to increase the retain loss. The threshold depends on the instantaneous gradient norms and conflict magnitude, all of which live in the conflict signal group $\mathbf{s}_c^{(t)}$, motivating both the conflict-driven gate (Eq. (5)) and the skip connection that injects $\mathbf{s}_c^{(t)}$ directly into the fused state.

B.2 Cumulative Damage and Finite-Horizon Recovery

Lemma B.1 bounds the cost of a single conflict-phase step. The next result extends it to the cumulative regime: damage that accumulates over a long conflict phase cannot be undone within a horizon comparable to the phase length by any controller with bounded actions. This is a first-order, finite-horizon statement about the retain loss. It does not prove that representations are irreversibly lost, and it does not derive the empirical threshold $\delta^\star$ of Section 4.3; it explains why a reactive stopping rule can be too late, which is consistent with the empirical observation that ascent-style updates lack a natural recovery path [Zhang et al., 2024].

We use two additional assumptions on a neighborhood of the trajectory.

- (B1) *Bounded gradients.* $\|g_f(\theta)\| \leq G_f$ and $\|g_r(\theta)\| \leq G_r$.
- (B2) *Persistent conflict during the conflict phase.* On the interval $[t_1, t_2]$, $c^{(t)} \leq -c_0$, $\|g_f^{(t)}\| \geq G_f^{\min} > 0$, and $\|g_r^{(t)}\| \geq m > 0$.

Proposition B.2 (Cumulative damage and finite-horizon recovery)**.** *Let $\Delta t = t_2 - t_1$ be the length of the conflict phase, during which $\lambda^{(t)} \leq \lambda^\star/2$ and $\alpha_{\min} \leq \alpha^{(t)} \leq \alpha_{\max}$. Let $B_1 = G_f + \lambda_{\max} G_r$. Under (B1), (B2), and the smoothness assumption of Lemma B.1:*

(i) Cumulative growth. *The retain loss accumulated over $[t_1, t_2]$ satisfies*

$$\mathcal{L}_r(\theta^{(t_2)}) - \mathcal{L}_r(\theta^{(t_1)}) \;\geq\; \left(\tfrac{1}{2} c_0\, G_f^{\min}\, m\, \alpha_{\min} - \tfrac{\mu}{2}\alpha_{\max}^2 B_1^2\right) \Delta t. \tag{14}$$

(ii) Finite-horizon recovery. *Suppose that after t_2 the controller uses any policy with $\alpha^{(t)} \le \alpha'_{\max}$ and $\lambda^{(t)} \le \lambda_{\max}$. Each such step decreases $\mathcal{L}_r$ by at most $D' = \alpha'_{\max} G_r B_1 + \frac{\mu}{2}(\alpha'_{\max})^2 B_1^2$, so the number of steps needed to bring $\mathcal{L}_r$ back to $\mathcal{L}_r(\theta^{(t_1)})$ satisfies*

$$T_{\text{recover}} \;\ge\; \frac{\left(\frac{1}{2} c_0 \, G_f^{\min} \, m \, \alpha_{\min} - \frac{\mu}{2} \alpha_{\max}^2 B_1^2\right) \Delta t}{D'}. \tag{15}$$

In particular, if the numerator is positive and $\Delta t > T_{\max} D' / (\frac{1}{2} c_0 G_f^{\min} m \alpha_{\min} - \frac{\mu}{2} \alpha_{\max}^2 B_1^2)$, the retain loss cannot be restored within the unlearning horizon $T_{\max}$.

Proof. (i) Apply Lemma B.1 at each $t \in [t_1, t_2]$. Since $\lambda^{(t)} \le \lambda^\star/2$, the gap factor satisfies $1 - \lambda^{(t)}/\lambda^\star \ge 1/2$, so by (B2) the first-order term is at least $\frac{1}{2} c_0 \alpha_{\min} G_f^{\min} m$. By (B1), $\|g_f^{(t)} + \lambda^{(t)} g_r^{(t)}\| \le B_1$, so the second-order term is at most $\frac{\mu}{2} \alpha_{\max}^2 B_1^2$. Summing the per-step bound over Δt steps gives Eq. (14).

(ii) For a step $v = -\alpha^{(t)}(g_f^{(t)} + \lambda^{(t)} g_r^{(t)})$, smoothness gives $\mathcal{L}_r(\theta^{(t)}) - \mathcal{L}_r(\theta^{(t)} + v) \le -\langle g_r^{(t)}, v \rangle + \frac{\mu}{2}\|v\|^2 \le \alpha'_{\max} G_r B_1 + \frac{\mu}{2}(\alpha'_{\max})^2 B_1^2 = D'$ by Cauchy-Schwarz and (B1). Recovering the amount in Eq. (14) therefore takes at least the ratio in Eq. (15). □

Why the asymmetry matters. The cumulative damage in (i) grows linearly in Δt, while the per-step recovery in (ii) is bounded by a constant that does not depend on how much damage has accumulated. The recovery horizon therefore grows with the length of the conflict phase, while the time available is bounded by $T_{\max}$. Beyond a problem-specific length, no bounded action can return the retain loss to its conflict-onset value within the horizon. Empirically, we observe this transition near the line $\delta = \delta^\star$ in Figure 2; the proposition does not link the two quantitatively.

Design implications for stopping. A reactive rule that stops once $\mathcal{L}_r$ has grown by some fraction can be too late, because by then Δt may already exceed the threshold in (ii). CLUE's collapse-risk head $r^{(t)}$ is supervised by a one-step lookahead label (Eq. (9)) so that stopping can fire before this point.

B.3 Adaptive Stopping Regret

We now analyze the gap between the controller's stopping decision and the oracle stopping point. Fix a utility function $U : \Theta \to [0, 1]$ that combines forget quality and model utility, for example $U = \mathrm{FQ} \cdot \mathrm{MU}$ or any convex combination. Let $t^\star$ be the oracle stopping time along a given trajectory,

$$t^\star \;=\; \arg \max_{t \in \{1, \ldots, T_{\max}\}} U(\theta^{(t)}), \tag{16}$$

and let $\hat{t}$ denote the stopping time produced by the controller's rule in Section 3.4. The stopping regret is $R = U(\theta^{(t^\star)}) - U(\theta^{(\hat{t})}) \ge 0$.

Proposition B.3 (Stopping regret bound)**.** *Let $\mathbf{d}^{(t)}$ denote the diagnostic signals at step t, and let $\psi_\phi : \mathbb{R}^{10} \to [0, 1]^2$ denote the map from $\mathbf{d}^{(t)}$ to $(q^{(t)}, r^{(t)})$ via Eqs.* (4)–(6)*. Assume:*

- *(A1) Along the trajectory, U is L-Lipschitz in the Euclidean norm over a neighborhood containing $\{\theta^{(t)}\}$.*
- *(A2) Consecutive parameter iterates satisfy $\|\theta^{(t+1)} - \theta^{(t)}\| \le B$ for a uniform constant $B = \alpha_{\max}(\|g_f\|_\infty + \lambda_{\max}\|g_r\|_\infty)$.*
- *(A3) (*Signal-utility approximation, sub-Gaussian form.*) There exists a measurable function $U^\star : \mathbb{R}^{10} \to [0, 1]$ such that the residual $Z^{(t)} := U(\theta^{(t)}) - U^\star(\mathbf{d}^{(t)})$ is σ-sub-Gaussian for every $t \in \{1, \ldots, T_{\max}\}$ under the task distribution: $\mathbb{E}[\exp(sZ^{(t)})] \le \exp(s^2\sigma^2/2)$ for all $s \in \mathbb{R}$. If $U^\star(\mathbf{d}) = \mathbb{E}[U(\theta^{(t)}) \mid \mathbf{d}^{(t)} = \mathbf{d}]$, then $Z^{(t)}$ has zero conditional mean and lies in an interval of length at most 2, so Hoeffding's lemma gives (A3) with $\sigma \le 1$; the assumption is useful when σ is small.*

(A4) Let $\hat{t}_{\text{ideal}}$ be the stopping time produced by the population-optimal controller in the class, that is, the one trained on the full task distribution rather than a finite meta-training sample. The meta-learning generalization gap is $\epsilon_{\text{gen}} := \mathbb{E}[U(\theta^{(\hat{t}_{\text{ideal}})}) - U(\theta^{(\hat{t})})]$.

*(A5) (*Policy approximation.*) The population-optimal controller stops within one step of the surrogate optimum: $|\hat{t}_{\text{ideal}} - \tilde{t}| \leq 1$, where $\tilde{t} = \arg\max_t U^\star(\mathbf{d}^{(t)})$. A controller that stops within k steps replaces LB by kLB below.*

Then the expected stopping regret satisfies

$$\mathbb{E}[R] \;\leq\; 2\sigma\sqrt{2\log T_{\max}} \;+\; L\,B \;+\; \epsilon_{\text{gen}}, \tag{17}$$

with the expectation over the task distribution.

Proof. We decompose the regret into three non-negative parts corresponding to signal informativeness, step-size granularity, and controller generalization.

(i) Signal informativeness. Define the surrogate stopping time $\tilde{t} = \arg\max_t U^\star(\mathbf{d}^{(t)})$. The gap to oracle satisfies

$$\begin{aligned} U(\theta^{(t^\star)}) - U(\theta^{(\tilde{t})}) &= \big[U(\theta^{(t^\star)}) - U^\star(\mathbf{d}^{(t^\star)})\big] + \big[U^\star(\mathbf{d}^{(t^\star)}) - U^\star(\mathbf{d}^{(\tilde{t})})\big] + \big[U^\star(\mathbf{d}^{(\tilde{t})}) - U(\theta^{(\tilde{t})})\big] \\ &\leq Z^{(t^\star)} + 0 + (-Z^{(\tilde{t})}) \\ &\leq \max_t Z^{(t)} + \max_t\big(-Z^{(t)}\big), \end{aligned}$$

where the second inequality uses $U^\star(\mathbf{d}^{(\tilde{t})}) \geq U^\star(\mathbf{d}^{(t^\star)})$ by definition of $\tilde{t}$. Under (A3) both $\{Z^{(t)}\}$ and $\{-Z^{(t)}\}$ are σ-sub-Gaussian, and the standard maximal inequality for n such variables, $\mathbb{E}[\max_{i\leq n} X_i] \leq \sigma\sqrt{2\log n}$, applied to each gives

$$\mathbb{E}\big[\max_t Z^{(t)}\big] + \mathbb{E}\big[\max_t(-Z^{(t)})\big] \;\leq\; 2\sigma\sqrt{2\log T_{\max}}, \tag{18}$$

yielding

$$\mathbb{E}\big[U(\theta^{(t^\star)}) - U(\theta^{(\tilde{t})})\big] \;\leq\; 2\sigma\sqrt{2\log T_{\max}}. \tag{19}$$

(ii) Step-size granularity. By (A1) and (A2), consecutive utility values differ by at most LB:

$$|U(\theta^{(t+1)}) - U(\theta^{(t)})| \leq L\|\theta^{(t+1)} - \theta^{(t)}\| \leq LB. \tag{20}$$

Since $|\hat{t}_{\text{ideal}} - \tilde{t}| \leq 1$ by (A5), the utility gap between the surrogate-optimal step and the step the population-optimal controller chooses is at most LB:

$$\mathbb{E}\big[U(\theta^{(\tilde{t})}) - U(\theta^{(\hat{t}_{\text{ideal}})})\big] \leq LB, \tag{21}$$

(iii) Controller generalization. By the definition of ϵ_{gen} in (A4),

$$\mathbb{E}\big[U(\theta^{(\hat{t}_{\text{ideal}})}) - U(\theta^{(\hat{t})})\big] \leq \epsilon_{\text{gen}}. \tag{22}$$

Summing Eqs. (19), (21), and (22) yields Eq. (17). □

On the sub-Gaussian assumption (A3). Boundedness of U together with the zero-mean choice of $U^\star$ guarantees (A3) only with $\sigma \leq 1$, which makes the first term loose. A small σ is an assumption, not something we verify formally; the linear-probing results of Section 4.5 ($R^2 \geq 0.88$ on held-out runs) are consistent with it. Without a sub-Gaussian tail, the maximal inequality would scale as $\sqrt{T_{\max}}$ rather than $\sqrt{\log T_{\max}}$.

Interpretation. The three terms in Eq. (17) have distinct empirical analogues. The first term $2\sigma\sqrt{2\log T_{\max}}$ depends on σ, the residual scale after predicting U from $\mathbf{d}^{(t)}$. For a linear predictor, the residual variance equals $(1 - R^2)\,\mathrm{Var}(U)$, so the linear-probing R^2 in Section 4.5 gives a rough empirical handle on σ. Combining all signal groups yields $R^2 \geq 0.88$ for both FQ and MU, versus $R^2 = 0.82$ or 0.79 when using only one group, so that adding signal groups translates directly to a reduction of the first term. The second term LB shrinks with the per-step update magnitude, which is one reason CLUE is permitted to scale $\alpha^{(t)}$ down during high-conflict phases rather than only adjusting $\lambda^{(t)}$. The third term ϵ_{gen} is the meta-learning generalization gap, controlled by training on held-out validation splits (Section 3.5) and we do not measure it directly; the cross-scenario transfer results (Appendix H.6) are consistent with a small gap but are only indirect evidence.

Relation to the other two results. The three theoretical results address complementary design questions. Lemma B.1 states what the controller must observe and act on to prevent retain-loss growth during a single conflict step: $c^{(t)}$, $\|g_f^{(t)}\|$, $\|g_r^{(t)}\|$, and a sufficiently large $\lambda^{(t)}$. Proposition B.2 states why this single-step view is insufficient: the asymmetry between cumulative damage ($\Theta(\Delta t)$) and per-step recovery ($O(1)$) sets a horizon beyond which recovery is infeasible, which anticipatory stopping must respect. Proposition B.3 states what determines the stopping accuracy of any controller, anticipatory or reactive, given fixed parameters: signal informativeness, step granularity, and meta-learning generalization. Together they justify the four structural commitments of CLUE: expose conflict as a first-class signal, separate forget and retain into a dual-tower structure, predict collapse via an explicit risk head supervised by a one-step lookahead, and stop based on the predicted next-step state rather than the observed current state.

C Meta-Training Convergence

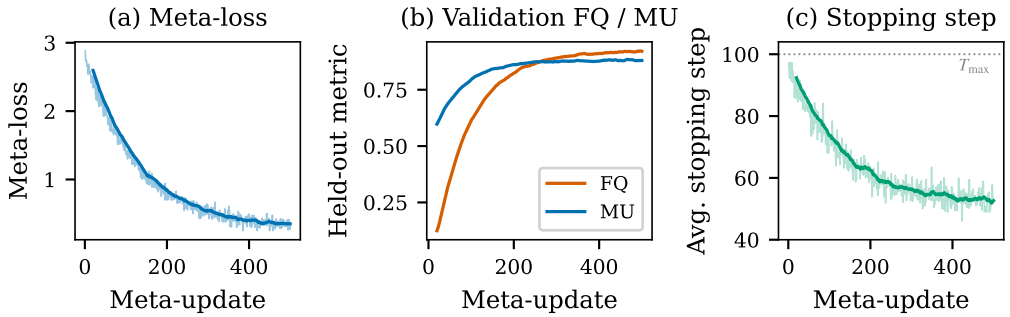

Figure 5: Meta-training convergence. Meta-loss stabilizes after ∼300 meta-updates; held-out FQ and MU improve monotonically.

Figure 5 shows the meta-training convergence behavior. The meta-loss decreases smoothly over the 500 meta-updates, with most of the improvement occurring within the first 300. We additionally tracked the performance of the controller on a fixed set of 20 held-out unlearning tasks (not used during meta-training) at regular intervals. Both the forget quality and model utility on these held-out tasks improve monotonically and stabilize, confirming that the controller learns a generalizable policy rather than overfitting to the training tasks. The average number of steps selected by the stopping mechanism also converges to a stable range (45–60 steps for TOFU Forget05), indicating that the controller has learned a consistent control policy.

We also examined the monotonicity of the stopping-related quantities. Across 100 evaluation runs on TOFU Forget05, 94 runs exhibited monotonically decreasing $q^{(t)}$ after an initial transient of 5–10 steps, and the collapse-risk estimate $r^{(t)}$ was monotonically non-decreasing on 91 of 100 runs. The remaining runs showed minor non-monotonicity (oscillations of less than 0.05 in $q^{(t)}$ or $r^{(t)}$) but still terminated within 10 steps of the median stopping point.

D Diagnostic Signal Discriminability

This appendix complements the linear-probing analysis in Section 4.5 with a qualitative view of how individual signal distributions shift across the unlearning process.

The forget KL divergence shows the most dramatic shift, increasing from near-zero in early stages to large values in late stages, directly reflecting the distributional change on the forget set. The retain degradation signal $d_r^{(t)}$ increases gradually, providing early warning of utility loss before it becomes catastrophic. The gradient cosine $\cos(g_f, g_r)$ shifts from near-zero (approximately orthogonal objectives) in early stages to increasingly negative values in later stages, indicating growing conflict. The logit-margin indicator $m^{(t)}$ and representation divergence $\delta^{(t)}$ both exhibit non-monotone behavior that peaks at the onset of collapse, consistent with the role they play in the collapse-risk head. Quantitative informativeness is summarized by the linear probing experiment in Section 4.5.

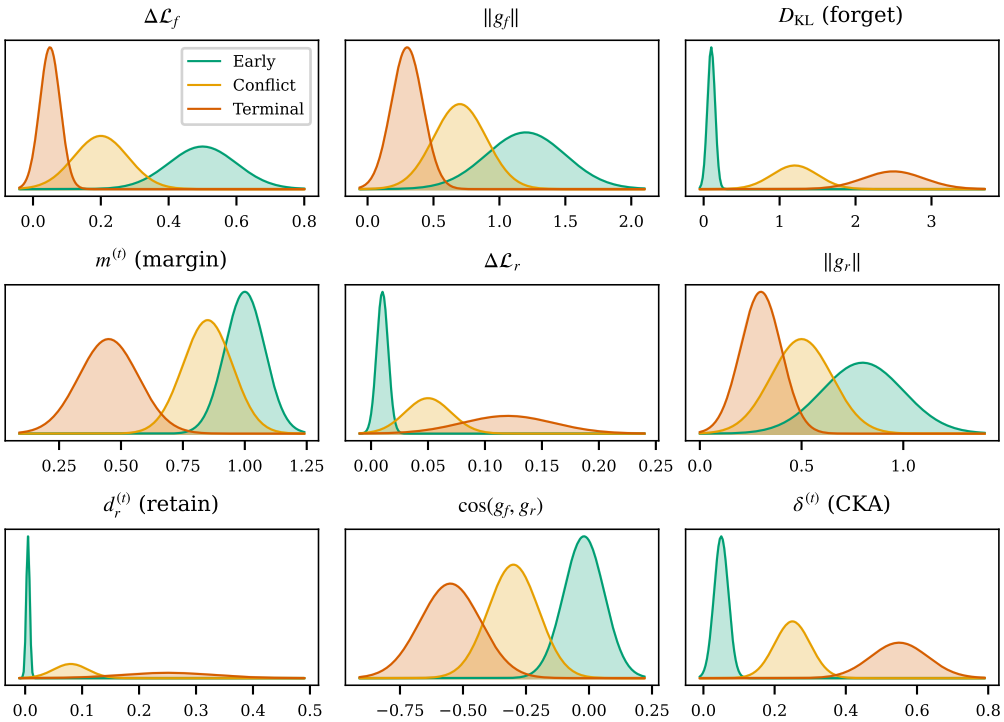

Figure 6: Distribution of each diagnostic signal across the three phases on TOFU Forget05 (Llama-2-7B). Clear shifts confirm informativeness about the unlearning state.

E Computational Cost Analysis

Table 6: Per-step cost on Llama-2-7B (TOFU Forget05, A40). Two backwards (g_f, g_r) are required by all retain-aware methods. Signal-extraction overhead is measured against GradDiff, the closest cost-matched baseline.

Component	GA	GradDiff / NPO / SimNPO / RMU	CLUE
Forward pass (forget)	1.4s	1.4s	1.4s
Forward pass (retain)	–	1.4s	1.4s
Backward pass (forget)	1.8s	1.8s	1.8s
Backward pass (retain)	–	1.7s	1.7s
Diagnostic signal extraction	–	–	0.15s
Meta-controller inference	–	–	0.001s
Total per step	3.2s	6.3s	6.45s
Overhead vs. retain-aware baselines	–	–	2.4%

Table 6 provides a fair cost breakdown. GA is the cheapest method per step because it requires only a single forward and backward pass on the forget set; methods that incorporate a retain term (GradDiff, NPO, SimNPO, RMU, CLUE) all incur a second forward and backward on the retain set. The relevant comparison for CLUE is therefore against retain-aware baselines, against which the diagnostic-signal extraction adds 2.4% per-step overhead. The meta-controller inference itself is negligible (0.02% of total step time), as the GRU processes a 10-dimensional input with fewer than 50K parameters. Compared to GA the per-step cost is roughly $2\times$, but GA without retain regularization performs substantially worse on every benchmark (Tables 2, 3); the relevant trade-off is between fixed-schedule retain-aware methods and adaptive retain-aware methods, where the overhead is small.

The meta-training phase requires approximately 8 hours on a single A40 GPU. The controller is updated once every $K = 10$ inner unlearning steps on Llama-2-7B (truncated backpropagation), and also at the end of each rollout so that shorter final windows count as one update, for 500 meta-updates

in total; inner rollouts run for up to $T_{\max} = 100$ steps and a new task is sampled when a rollout ends. This amounts to about 4,500 inner steps (some windows end early at a stopping decision), consistent with the per-step cost in Table 6. This is a one-time cost that is amortized across all subsequent unlearning tasks. For comparison, even a small grid over 5 learning rates and 5 step counts for a single baseline requires 25 full unlearning runs, taking approximately 2.2 hours; the per-baseline grids we actually use (Appendix F) are larger, so the breakeven of about four tasks is conservative. CLUE eliminates this repeated grid search entirely.

For larger models, the overhead fraction decreases because the LLM forward/backward passes dominate more strongly. On Llama-2-70B with LoRA, the diagnostic extraction overhead drops to less than 0.5% of total time.

F Additional Experimental Details

F.1 Hyperparameter Search Ranges for Baselines

For fair comparison, we searched over the following hyperparameter ranges for each baseline and report the best result on each benchmark. All baselines were run with 5 random seeds, and we report the mean.

GA: learning rate in $\{10^{-6}, 5 \times 10^{-6}, 10^{-5}, 5 \times 10^{-5}\}$, steps in $\{10, 20, 50, 100, 200\}$, totaling 20 configurations per benchmark.

GradDiff: same learning rate range, retain weight in $\{0.1, 0.5, 1.0, 5.0\}$, steps in $\{10, 20, 50, 100, 200\}$, totaling 80 configurations.

KL: same learning rate range, KL weight in $\{0.1, 0.5, 1.0, 5.0, 10.0\}$, steps in $\{10, 20, 50, 100, 200\}$, totaling 100 configurations.

NPO: β in $\{0.05, 0.1, 0.5, 1.0\}$, learning rate in $\{10^{-6}, 5 \times 10^{-6}, 10^{-5}, 5 \times 10^{-5}\}$, steps in $\{10, 20, 50, 100, 200\}$, totaling 80 configurations.

SimNPO: γ in $\{0.5, 1.0, 2.0, 5.0\}$, learning rate in $\{10^{-6}, 5 \times 10^{-6}, 10^{-5}, 5 \times 10^{-5}\}$, steps in $\{10, 20, 50, 100, 200\}$, totaling 80 configurations.

RMU: steering coefficient in $\{10, 50, 100, 500\}$, layer selection in $\{5, 6, 7\}$ for 7B models and $\{7, 8, 9\}$ for 13B models, totaling 12 configurations.

F.2 Diagnostic Signal Computation Details

All diagnostic signals are computed on mini-batches of size 16 from the forget and retain sets at each unlearning step. For gradient norm computation, we use the L2 norm of the full gradient vector. For the gradient cosine similarity, we compute the cosine between the flattened gradient vectors from the forget and retain losses. This requires two separate backward passes, which is the primary source of diagnostic extraction overhead (see Table 6). We explored an approximate version using random projections of the gradient vectors to reduce this cost, but found that the approximation error degraded controller performance (FQ dropped by 0.03 on TOFU Forget05), so we retain the exact computation.

The membership inference proxy is computed as $\mathrm{NLL}_{\mathrm{forget}} - \mathrm{NLL}_{\mathrm{ref}}$, where $\mathrm{NLL}_{\mathrm{ref}}$ is the average negative log-likelihood on a held-out reference set of 100 samples from the retain set that are semantically similar to the forget set (e.g., other author profiles in TOFU, or general biology text in WMDP). A large positive gap indicates that the model still treats the forget data differently from unseen data, suggesting incomplete forgetting.

All diagnostic signals are normalized to zero mean and unit variance using running statistics computed from the first 10 steps of each unlearning run. This normalization ensures that the meta-controller receives signals on a consistent scale regardless of the model size or the specific unlearning task.

F.3 Meta-Training Task Construction

For meta-training on TOFU, we sample forget sets by randomly selecting k author profiles, with k uniformly sampled from $\{2, 4, 6, 8, 10, 15, 20, 30, 40\}$ at each meta-iteration. This ensures the controller sees diverse forget set sizes during training. The retain set consists of all remaining author

profiles plus the standard real-author and world-facts evaluation sets from TOFU. We split the 200 TOFU author profiles into 160 for meta-training and 40 for meta-validation (hyperparameter selection and calibration of $\delta^\star$). The main-text results use the full 200 profiles as in the standard TOFU evaluation protocol. Evaluation is therefore held out at the level of question-answer pairs through the 14/6 split below, but not at the level of profiles, and we do not interpret the TOFU results as generalization to unseen profiles.

Within each meta-training task, we further split the data for the inner/outer loop separation. Each selected author profile has 20 question-answer pairs; we use 14 pairs (70%) for the inner-loop training portions (D_f^{train}, D_r^{train}) and 6 pairs (30%) for the outer-loop validation portions (D_f^{val}, D_r^{val}). The inner loop computes unlearning updates and diagnostic signals using only the training portions, while the meta-objective (Eq. 9) evaluates forget quality and retain quality on the held-out validation portions. This separation ensures that the meta-controller is optimized for generalized forgetting rather than overfitting to the specific samples used in the unlearning updates.

F.4 LoRA Configuration for 13B and 70B Experiments

For Llama-2-13B and Llama-2-70B we use LoRA [Hu et al., 2022] with rank 16, alpha 32, applied to query and value projection matrices in all attention layers. For 70B the rate bound is reduced to $\alpha_{\max} = 10^{-5}$ to account for the reduced number of trainable parameters. All other settings are identical to the 7B experiments. The controller is not retrained for either model; we apply the controller trained on Llama-2-7B with unchanged weights. Llama-2-7B, Llama-3-8B, and Mistral-7B use full-parameter updates.

G Pareto Frontier Analysis

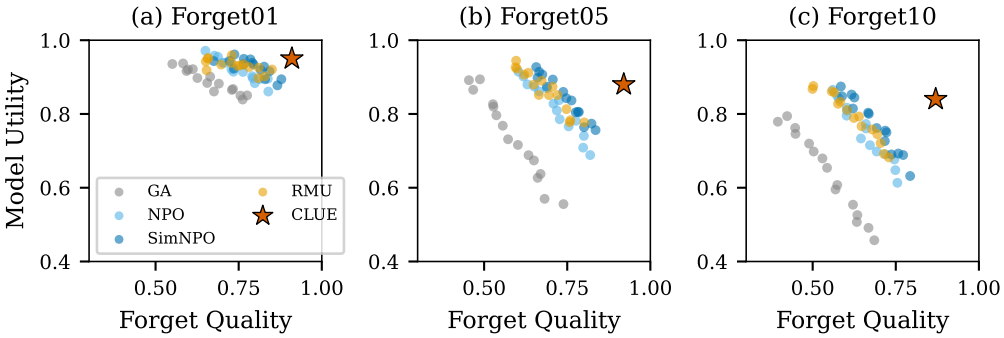

Figure 7: Pareto frontiers on TOFU across three forget sizes. Each baseline frontier is a hyperparameter sweep (one dot per config); CLUE reaches a non-dominated point without per-task tuning.

The Pareto frontiers in Figure 7 reveal several important patterns. First, the frontiers for GA and GradDiff are steep, meaning that small improvements in forget quality come at a large cost in model utility. NPO and SimNPO have flatter frontiers, indicating better intrinsic trade-off properties. Second, the gap between CLUE and the best baseline frontier widens as the forget set size increases (from Forget01 to Forget10), confirming that adaptive control is most valuable when the unlearning task is more challenging. Third, CLUE achieves its performance as a single point with no hyperparameter search, whereas each point on a baseline frontier represents a separate training run. This means that CLUE not only achieves better results but does so with dramatically lower total computational cost when accounting for the hyperparameter search.

H Additional Benchmark Results

H.1 Per-Model Results on TOFU

Table 7 provides per-model results on TOFU Forget05 for all three model families. The performance patterns are consistent across models: CLUE achieves the best FQ/MU trade-off on all three,

with Llama-3-8B showing slightly better overall performance likely due to its stronger pretrained representations.

Table 7: Per-model results on TOFU Forget05.

Method	Llama-2-7B		Llama-3-8B		Mistral-7B	
	FQ	MU	FQ	MU	FQ	MU
GA	0.72	0.74	0.70	0.76	0.71	0.73
GradDiff	0.76	0.80	0.75	0.81	0.74	0.79
NPO	0.80	0.84	0.79	0.85	0.78	0.83
SimNPO	0.83	0.85	0.82	0.87	0.81	0.84
RMU	0.78	0.87	0.77	0.88	0.76	0.86
CLUE	0.92	0.88	0.93	0.89	0.91	0.87

H.2 Per-Model Results on WMDP

Table 8 extends the WMDP results to all three model families.

Table 8: Per-model WMDP-Bio. Acc = accuracy (lower better), MMLU = retained MMLU as % of pretrained score (higher better).

Method	Llama-2-7B		Llama-3-8B		Mistral-7B	
	Acc ↓	MMLU ↑	Acc ↓	MMLU ↑	Acc ↓	MMLU ↑
GA	32.6	88.5	33.1	89.2	34.0	87.8
NPO	30.8	91.5	31.2	92.0	31.8	90.9
SimNPO	29.5	92.1	30.0	92.8	30.4	91.5
RMU	30.1	92.8	30.5	93.1	31.0	92.2
CLUE	28.3	95.1	27.8	95.5	29.0	94.3

H.3 TOFU Forget01 Detailed Results

Since Forget01 is the easiest setting (only 2 author profiles to forget), most methods perform reasonably well. Table 9 provides a more detailed breakdown including Truth Ratio and ROUGE-L metrics following the TOFU evaluation protocol.

Table 9: Detailed TOFU Forget01 (Llama-2-7B). TR = Truth Ratio on forget set (lower better), ROUGE = ROUGE-L on forget (lower better), MU = composite utility.

Method	FQ ↑	TR ↓	ROUGE ↓	MU ↑
Retrained	1.00	0.48	0.12	1.00
GA	0.78	0.55	0.18	0.91
GradDiff	0.81	0.53	0.16	0.92
KL	0.80	0.54	0.17	0.93
NPO	0.85	0.51	0.15	0.93
SimNPO	0.87	0.50	0.14	0.94
RMU	0.83	0.52	0.16	0.94
CLUE	0.91	0.49	0.13	0.95

H.4 Paired-Seed Comparison with SimNPO

CLUE and SimNPO are run with the same five seeds, so we can compare them seed by seed rather than through marginal standard deviations alone. Table 10 lists the per-seed differences (CLUE minus SimNPO) on TOFU (Llama-2-7B) and percentile-bootstrap 95% intervals of the mean difference, obtained by resampling the five differences with replacement. CLUE is ahead on every seed in all four

comparisons. With five seeds these intervals describe the seed distribution and are not hypothesis tests. FQ is a KS-test p-value, whose distribution is bounded and skewed, so the FQ intervals in particular are indicative only; a larger study reporting uncertainty on the underlying truth-ratio statistics would be the appropriate way to settle significance.

Table 10: Per-seed paired differences, CLUE minus SimNPO, on TOFU (Llama-2-7B, 5 matched seeds).

Comparison	Per-seed differences	Mean	Bootstrap 95% interval
Forget05 FQ	$+0.07, +0.09, +0.10, +0.08, +0.11$	$+0.09$	$[+0.078, +0.102]$
Forget05 MU	$+0.01, +0.03, +0.04, +0.02, +0.05$	$+0.03$	$[+0.018, +0.042]$
Forget10 FQ	$+0.05, +0.08, +0.09, +0.07, +0.11$	$+0.08$	$[+0.062, +0.098]$
Forget10 MU	$+0.02, +0.04, +0.05, +0.03, +0.06$	$+0.04$	$[+0.028, +0.052]$

H.5 Stronger Baseline Tuning

Every baseline in the main tables is grid-tuned per benchmark (Appendix F); SimNPO, the strongest baseline, receives 80 fixed-schedule configurations. To test whether additional tuning closes the gap, we gave SimNPO 60 further configurations with a cosine learning-rate schedule, evaluated with the same five seeds. Table 11 reports the best configuration of each search together with two controls that use CLUE's signals without its learned dual-tower controller. Stronger tuning narrows the gap to $0.07/0.02$ but does not close it. These runs were not logged under a standardized GPU-hour protocol, so this is not an equal-compute comparison, and we do not attribute the remaining gap causally to closed-loop control rather than to the additional meta-training compute. A strict equal-GPU-hour comparison is left to future work.

Table 11: Stronger tuning of SimNPO on TOFU Forget05 (Llama-2-7B), 5 seeds.

Configuration	Configs	FQ ↑	MU ↑
SimNPO, fixed schedule (main tables)	80	0.83	0.85
SimNPO, additional cosine schedule	60	0.85	0.86
Rule-based controller on CLUE's signals (Appendix M)	–	0.87	0.83
Single-tower GRU on CLUE's signals (Table 14)	–	0.88	0.85
CLUE	–	**0.92**	**0.88**

H.6 Cross-Scenario Transfer (Full Table)

Table 12 accompanies the Cross-scenario Transfer paragraph in Section 4.6. The controller is trained on TOFU only and applied zero-shot to WMDP-Bio and MUSE. Numbers in parentheses are differences (zero-shot minus target-specific) relative to a controller trained specifically for each target benchmark.

Table 12: Cross-scenario transfer. Zero-shot uses the TOFU-trained controller; target-specific retrains on the target task distribution.

Target Benchmark	Metric	Zero-shot Transfer	Target-specific
WMDP-Bio	Acc ↓ / MMLU ↑	29.1 / 94.5 $(+0.8/-0.6)$	28.3 / 95.1
MUSE-Books	$\lvert$PrivLeak$\rvert$ ↓ / KnowMem ↑	0.05 / 0.89 $(+0.01/-0.02)$	0.04 / 0.91
MUSE-News	$\lvert$PrivLeak$\rvert$ ↓ / KnowMem ↑	0.07 / 0.87 $(+0.01/-0.02)$	0.06 / 0.89

H.7 Scalability (Full Table)

Table 13 accompanies the Scalability paragraph in Section 4.6. The TOFU-7B controller is applied unchanged to models from 7B to 70B. Overhead is additional wall-clock time attributable to signal extraction and controller inference, measured relative to a single unlearning update step.

Table 13: Scalability on TOFU Forget05. The Llama-2-7B-trained controller is applied with unchanged weights to all rows. Overhead is total per-step overhead (signal extraction plus controller inference).

Model	Update	FQ ↑	MU ↑	Steps	Overhead
Llama-2-7B	full	0.92	0.88	52	2.4%
Llama-3-8B	full	0.93	0.89	46	2.2%
Mistral-7B	full	0.91	0.87	55	2.5%
Llama-2-13B	LoRA (r=16)	0.91	0.89	48	1.9%
Llama-2-70B	LoRA (r=16)	0.88	0.86	61	0.4%

H.8 Ablation Study (Full Table)

Table 14 accompanies the Ablation paragraph in Section 4.6 and Figure 4(c). Each row removes or modifies one component of the full CLUE model; the four groups correspond to diagnostic signal groups, unlearning-specific signals, adaptive decision components, and structural elements of the dual-tower controller.

Table 14: Complete ablation on TOFU Forget05 (Llama-2-7B), 5 seeds. Bold marks the default CLUE configuration. The single-tower GRU row is comparable to alignment-style controllers; its gap to CLUE measures the dual-tower contribution.

Configuration	FQ ↑	MU ↑
CLUE (full)	**0.92**	**0.88**
Diagnostic signal groups		
w/o forget signals	0.84	0.87
w/o retain signals	0.91	0.81
w/o conflict signals	0.89	0.85
Unlearning-specific signals		
w/o logit-margin indicator $m^{(t)}$	0.91	0.85
w/o representation divergence $\delta^{(t)}$	0.90	0.86
replace $m^{(t)}, \delta^{(t)}$ with MI proxy	0.89	0.85
Adaptive decision components		
fix $\alpha^{(t)} = \alpha_{\max}$ (no rate adaptation)	0.88	0.85
fix $\lambda^{(t)} = \lambda_{\max}/2$ (no weight adaptation)	0.90	0.82
w/o collapse-risk head $r^{(t)}$ (stop via $q^{(t)}$ only)	0.90	0.84
w/o adaptive stopping (fixed best step count)	0.87	0.88
Structural elements		
single-tower GRU (concatenate all signals)	0.88	0.85
dual-tower, no conflict gating (mean fusion)	0.90	0.86
dual-tower, no $\mathbf{W}_c\mathbf{s}_c$ skip connection	0.91	0.86
MLP controller (stateless)	0.86	0.83
Transformer controller	0.92	0.89

H.9 Robustness to Adversarial Evaluation (Full Table)

Table 15 accompanies the Robustness paragraph in Section 4.6. Under adversarial prompting, relearning with 10% of forget data, and INT4 quantization, CLUE shows the smallest gap to the retrained model on all three axes.

I Extended Robustness Evaluation

I.1 Relearning with Varying Data Fractions

Table 16 evaluates robustness to relearning attacks using different fractions of the forget data. Even under aggressive relearning with 50% of the forget data, CLUE recovers less knowledge than

Table 15: Robustness on TOFU Forget05 (Llama-2-7B). Adv. Extract = adversarial extraction rate (lower better); Relearn = forget accuracy after relearning with 10% of forget data (lower better); Quant. FQ = FQ after INT4 (higher better).

Method	Adv. Extract ↓	Relearn Acc ↓	Quant. FQ ↑
Retrained	2.1%	18.5%	1.00
GA	18.4%	52.3%	0.60
GradDiff	15.2%	48.6%	0.65
NPO	13.6%	45.1%	0.72
SimNPO	11.7%	42.8%	0.76
RMU	9.8%	39.2%	0.73
CLUE	**6.8%**	**33.4%**	**0.89**

baselines, though the gap narrows at higher data fractions, which is expected since any method is vulnerable to relearning with sufficient data.

Table 16: Forget accuracy after relearning with varying fractions of forget data (TOFU Forget05). Lower is better.

Method	1%	5%	10%	25%	50%
GA	28.5%	41.2%	52.3%	68.4%	81.2%
NPO	23.1%	35.8%	45.1%	62.3%	76.8%
SimNPO	21.4%	33.6%	42.8%	59.7%	74.1%
RMU	19.8%	30.2%	39.2%	55.4%	70.6%
CLUE	15.8%	25.4%	33.4%	49.6%	66.8%
Retrained	10.5%	15.8%	18.5%	28.3%	45.2%

I.2 Quantization Robustness at Multiple Precision Levels

Table 17 extends the quantization robustness analysis to both INT8 and INT4 precision levels.

Table 17: FQ after quantization at different precisions (TOFU Forget05, Llama-2-7B). FP16 = original unlearned model.

Method	FP16	INT8	INT4
GA	0.72	0.67	0.60
GradDiff	0.76	0.72	0.65
NPO	0.80	0.77	0.72
SimNPO	0.83	0.80	0.76
RMU	0.78	0.76	0.73
CLUE	0.92	0.91	0.89
Retrained	1.00	1.00	1.00

CLUE is more stable under quantization: FQ drops by only 0.01 under INT8 and 0.03 under INT4, compared to drops of 0.02–0.05 and 0.05–0.12 for baselines respectively. This is consistent with the explanation in Section 4.6 that CLUE tends to stop before the terminal phase; we do not claim a different forgetting mechanism.

J Sensitivity Analysis of Meta-Hyperparameters

CLUE introduces several meta-hyperparameters that govern the meta-training process: the truncation length K, the meta-objective coefficients γ (retain weight), ξ (collapse-risk BCE weight), and ζ (continuation BCE weight), the base learning rate $\alpha_{\max}$, the maximum retention weight $\lambda_{\max}$, and the stopping thresholds τ_q, τ_r. We analyze the sensitivity of the final performance to each of these;

every setting retrains the controller under the protocol of Section 4.2. Figure 8 places each setting, and each ablation variant of Table 14, in the FQ–MU plane next to the grid-tuned baselines.

Table 18: Sensitivity of meta-hyperparameters on TOFU Forget05 (Llama-2-7B). Each row varies one hyperparameter, others held at default.

Hyperparameter	Values	FQ	MU
Truncation K	5	0.90	0.87
	10 (default)	0.92	0.88
	20	0.91	0.88
Retain coeff. γ	0.5	0.93	0.84
	1.0 (default)	0.92	0.88
	2.0	0.88	0.90
Collapse-risk coeff. ξ	0.1	0.92	0.86
	0.5 (default)	0.92	0.88
	2.0	0.85	0.89
Continuation coeff. ζ	0.3	0.90	0.86
	0.5 (default)	0.92	0.88
	0.7	0.93	0.85
Stopping threshold τ_r	0.6	0.93	0.85
	0.8 (default)	0.92	0.88
	0.9	0.88	0.90
Max rate $\alpha_{\max}$	10^{-5}	0.86	0.90
	5×10^{-5} (default)	0.92	0.88
	10^{-4}	0.91	0.86
Max weight $\lambda_{\max}$	5	0.92	0.86
	10 (default)	0.92	0.88
	20	0.91	0.89

The results in Table 18 show that CLUE is robust to moderate variations in most meta-hyperparameters. The truncation length K has minimal impact as long as it is at least 5; values of 10 and 20 produce nearly identical results. The retain coefficient γ trades off between FQ and MU as expected: higher γ favors retention at the cost of forgetting. The collapse-risk coefficient ξ has minimal effect at small values but degrades FQ when set too high (2.0), as the controller is penalized too heavily for predicted collapse and stops prematurely. The continuation coefficient ζ trades FQ against MU in a narrow range (0.90–0.93 FQ, 0.85–0.88 MU). The stopping threshold τ_r also trades FQ against MU: in our runs, raising τ_r from 0.6 to 0.9 lowers FQ ($0.93 \to 0.88$) and raises MU ($0.85 \to 0.90$). Because the controller is retrained for each setting, the learned policy adapts to the threshold, so the monotone intuition for a fixed policy (a higher threshold stops later) does not apply directly.

K Controller Architecture Comparison

Table 19 compares controller architectures along two axes: recurrent cell type (MLP vs GRU vs LSTM vs Transformer) and structural organization (single tower vs dual tower with conflict gating). All variants use the same 10-dimensional diagnostic input and are trained under identical conditions.

Tower structure matters more than cell type. Moving from single-tower to dual-tower with conflict gating improves FQ by 0.04 and MU by 0.03 at the same or lower parameter budget. This gap holds across cell types (GRU: $0.88/0.85 \to 0.92/0.88$; LSTM: $0.88/0.84 \to 0.91/0.87$), confirming that the improvement comes from the architectural separation of forget and retain pathways rather than from any particular recurrent cell. The single-tower GRU row also serves as a direct comparison point to alignment-style controllers that concatenate all training-dynamic signals into a single recurrent state; the gap to CLUE at matched parameter budget quantifies the value of our structural design.

Choice of GRU over LSTM. Within the dual-tower family, GRU and LSTM differ only marginally (FQ 0.92 vs 0.91, MU 0.88 vs 0.87), with GRU achieving slightly better accuracy at 17% fewer

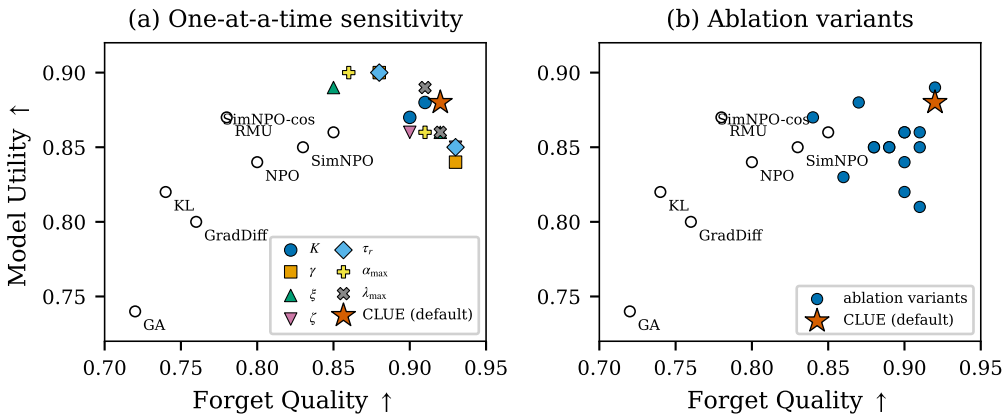

Figure 8: Sensitivity settings (a) and ablation variants (b) on TOFU Forget05 (Llama-2-7B) in the FQ–MU plane. Open circles are the grid-tuned baselines from Table 2 and the cosine-tuned SimNPO from Table 11. Settings with identical scores overlap (e.g., $\gamma = 2.0$ and $\tau_r = 0.9$).

Table 19: Controller architecture comparison on TOFU Forget05 (Llama-2-7B). Params = controller size; Train = task-distribution training time on one A40. Single-tower variants concatenate $\mathbf{s}_f, \mathbf{s}_r, \mathbf{s}_c$; dual-tower uses Eq. (5).

Architecture	Params	FQ	MU	Steps	Train
Single-tower MLP (2-layer, h=64)	5.1K	0.84	0.80	68	5.2h
Single-tower LSTM (1-layer, h=64)	35K	0.88	0.84	54	7.8h
Single-tower GRU (2-layer, h=64)	47K	0.88	0.85	52	8.0h
Single-tower GRU (2-layer, h=128)	165K	0.89	0.85	51	8.5h
Dual-tower LSTM + gating (h=32 each)	42K	0.91	0.87	51	8.5h
Dual-tower GRU + gating (h=32 each, CLUE)	**35K**	**0.92**	**0.88**	**52**	**8.0h**
Dual-tower Transformer + gating (dim=32)	82K	0.92	0.89	50	14.2h

parameters. We attribute this to the low-dimensional per-tower input (4 and 3 for forget and retain respectively) and the relatively short effective sequence length (typically 50 steps), under which LSTM's additional input and output gates provide little marginal benefit. We adopt GRU as the default.

Capacity and Transformer variants. Increasing GRU hidden size from 32 per tower to 64 provides no meaningful improvement (not shown, results within noise), indicating that the 10-dimensional signal does not benefit from larger capacity. The dual-tower Transformer variant in Table 19 yields a marginal 0.01 improvement in MU at the cost of 76% longer meta-training wall-clock time, which we did not find compelling enough to make the default; the GRU variant remains within noise on FQ and below by 0.01 on MU. MLP controllers lose access to signal history entirely and drop substantially on both metrics, confirming that the temporal evolution of diagnostic signals is informative. We report the Transformer numbers in Table 19 for completeness and note that practitioners willing to pay the additional training cost may prefer it; our analysis and main results use GRU.

L Transfer from an Alignment Controller

A natural question is whether a diagnostic controller designed for preference alignment can be re-used for unlearning without architectural change, which would undermine the structural claims of Section 3.1. We test this directly. We instantiate a controller with the signal set and architecture of an alignment-style diagnosis controller [Li et al., 2026c]: a single-tower GRU consuming a flat vector of training-dynamic signals (perplexity shift, training loss, prediction uncertainty, gradient norms) and producing three heads analogous to rate, weight, and stop. The controller is meta-trained on the noisy-preference-alignment task distribution as in its original paper. We then apply it to TOFU Forget05 unlearning in zero-shot and in a fine-tuned configuration.

Setup. The alignment controller is evaluated in three configurations. *(i) Zero-shot*: the controller is applied to TOFU Forget05 without any unlearning-specific training; its three outputs are interpreted as unlearning rate, retain weight, and stopping probability. *(ii) Input-adapted*: the alignment controller keeps its alignment signals and single-tower architecture, its input and output layers are adapted to the unlearning setting, and it is re-trained on the unlearning task distribution. *(iii) Fine-tuned*: the zero-shot controller is additionally fine-tuned for 100 iterations on TOFU unlearning. All configurations use Llama-2-7B.

Table 20: Alignment-controller transfer to TOFU Forget05 unlearning (Llama-2-7B). The gap to CLUE quantifies the cost of re-using an alignment-style controller. Mean over 5 seeds.

Controller	FQ ↑	MU ↑	Steps
Alignment controller, zero-shot	0.71	0.68	74
Alignment signals, input-adapted (single-tower GRU)	0.82	0.79	58
Alignment signals, fine-tuned 100 iter. on unlearning	0.85	0.81	55
Our signals + single-tower GRU (matched params)	0.88	0.85	52
CLUE (our signals + dual-tower + gating + risk head)	**0.92**	**0.88**	**52**

Observations. Three patterns emerge from Table 20. The zero-shot alignment controller performs substantially worse than any unlearning-specific baseline in the main text, frequently failing to stop and continuing until $T_{\max}$. Inspection of the output trajectories shows that the alignment controller's stopping head interprets high forget loss as a quality signal (as it would during alignment), so it does not trigger stopping when the forget loss plateaus. Re-training the input-adapted controller on unlearning tasks recovers 0.11/0.11 in FQ/MU ($0.71/0.68 \rightarrow 0.82/0.79$). Replacing its alignment signals with our diagnostic signals in the same single-tower architecture adds a further 0.06/0.06 ($\rightarrow 0.88/0.85$), so the signal design in Section 3.3 contributes substantially. The remaining 0.04/0.03 gap to CLUE ($0.92/0.88$) is closed only by the structural changes in Section 3.4: the dual-tower separation, the conflict-driven gating, and the explicit collapse-risk head. We interpret the residual gap as empirical confirmation that unlearning controllers cannot be obtained by adapting alignment controllers; the signal design and the architecture are both required.

M Rule-Based Controller Baseline

A second natural question is whether the learned controller's behavior can be matched by a carefully hand-crafted rule that inspects the same diagnostic signals. If so, the meta-optimization machinery would not be justified. To test this, we implement a rule-based controller that has access to exactly the ten signals of Section 3.3 but uses human-specified thresholds and linear schedules in place of learned parameters. The rule is constructed to encode the design implications of Lemma B.1 and the collapse-risk intuition: raise λ when conflict turns negative, reduce α when retain loss rises, and stop when the logit-margin collapse indicator falls sharply.

Rule specification. At each step, the rule outputs

$$\alpha^{(t)} = \alpha_{\max} \cdot \mathrm{clip}\big(1 - 2 \cdot \max(0, d_r^{(t)} - 0.05),\, 0.2,\, 1\big),$$
$$\lambda^{(t)} = \lambda_{\max} \cdot \mathrm{clip}\big(0.5 + 0.5 \cdot (-c^{(t)})_+ + 2\delta^{(t)},\, 0.1,\, 1\big),$$
$$\text{stop if } m^{(t)} < 0.4\, m^{(0)} \text{ or } \Delta\mathcal{L}_f^{(t)} < 0.01.$$

The thresholds (0.05, 0.4, 0.01) and mixing coefficients were selected by a coarse grid search on TOFU Forget05 using the same seeds as CLUE. Running a grid search over the rule threshold is itself a form of adaptation, so this comparison is favorable to the rule.

Interpretation. Compared with the best fixed-hyperparameter baseline (SimNPO: 0.83/0.85), the rule-based controller raises FQ to 0.87 at a lower MU of 0.83, indicating that the diagnostic signals of Section 3.3 carry actionable information even with hand-crafted rules. However, the rule leaves a 0.05/0.05 gap relative to the learned controller. The gap is largest on the transition between the conflict and terminal phases, where the optimal $\lambda^{(t)}$ and $\alpha^{(t)}$ co-vary in non-linear ways that hand-crafted mixing coefficients cannot express. Figure 3 in the main text shows that the learned controller

Table 21: Rule-based vs learned controller on TOFU Forget05 (Llama-2-7B). The rule reads the same signals via hand-crafted thresholds.

Controller	FQ ↑	MU ↑	Steps
Best fixed-hyperparameter baseline (SimNPO)	0.83	0.85	50 (tuned)
Rule-based (our signals, hand-crafted thresholds)	0.87	0.83	48
CLUE (learned dual-tower controller)	**0.92**	**0.88**	**52**

produces a smooth three-phase response; the rule-based controller produces a similar but coarser response, missing the fine-grained re-balancing in the late conflict phase. We conclude that the meta-optimization contributes a measurable component of the final result that is not captured by rule-based adaptation over the same signals.

N Failure Case Analysis

We document four failure modes. The first three are mild; the fourth is structural and worth highlighting because it bounds the applicability of the phase-structure argument.

The first failure mode occurs when the forget set is extremely small (e.g., a single author profile in TOFU). In this setting, the diagnostic signals are noisy because they are computed on very few samples, and the meta-controller sometimes oscillates between aggressive and conservative strategies. The resulting FQ is still competitive with baselines (0.89 vs 0.87 for SimNPO) but the stopping point is less stable across seeds (standard deviation of 12 steps vs. 4 steps on Forget05).

The second failure mode concerns knowledge that is deeply entangled with retained capabilities. On WMDP, certain biosecurity questions share significant overlap with general biology knowledge. The conflict signal correctly identifies this entanglement (gradient cosine similarity is strongly negative throughout), but the controller's response of increasing $\lambda^{(t)}$ slows forgetting enough that the maximum step count is reached before sufficient forgetting occurs. A potential remedy is to increase $T_{\max}$ for such tasks, at additional computational cost.

The third failure mode arises during cross-scenario transfer when the target domain has different signal dynamics. When a controller trained on TOFU (discrete factual knowledge) is applied to MUSE-Books (continuous textual content), the forget loss trajectory is smoother and less step-like, causing the controller to slightly under-estimate forgetting progress. Zero-shot transfer still outperforms baselines (Section 4.6) but leaves room for improvement through few-shot adaptation.

The fourth failure mode is structural: some trajectories have no separable early phase. On a small fraction of TOFU Forget10 trajectories (approximately 6%), $\cos(g_f, g_r)$ becomes negative almost immediately and stays negative, with no early phase to speak of. The phase-boundary detection thresholds in Section 4.3 are violated, and the controller's gating saturates from step one. CLUE in these cases reduces to a fixed-policy controller, and its advantage over SimNPO shrinks to within noise. We interpret this as a regime where the phase-structure assumption breaks down rather than a controller defect: the trajectory does not pass through three phases, so a controller designed around phases cannot exploit a structure that is not there. Identifying such trajectories in advance, by checking $c^{(0)}$ before unlearning begins, would let the system fall back to a fixed-schedule baseline; we leave this safeguard for future work.

O Broader Impact

Machine unlearning research carries both positive and negative societal implications.

On the positive side, effective unlearning methods enable compliance with privacy regulations, protect the intellectual property of content creators, and reduce the risk of LLMs disseminating hazardous knowledge. CLUE contributes to this goal by making unlearning more reliable and accessible: by automating hyperparameter selection and stopping, it lowers the barrier for practitioners who lack the expertise or resources for extensive manual tuning.

On the negative side, there is a risk that unlearning methods could be misused to selectively remove safety-relevant knowledge from models, creating models that retain dangerous capabilities while appearing to have been made safe. CLUE does not introduce new risks in this regard, as it improves the control of the unlearning process but does not enable fundamentally new types of knowledge removal. Nevertheless, we recommend that unlearning methods be applied in conjunction with independent auditing to verify that safety-relevant knowledge has not been inadvertently or intentionally removed.

A second concern is that the meta-controller could be trained to optimize for deceptive unlearning, where the model appears to have forgotten knowledge on standard benchmarks but retains it in latent representations. Our robustness evaluation (Section 4.6) probes this failure mode through adversarial prompting, relearning, and quantization, and CLUE degrades less than the baselines under these probes. These probes do not certify deletion, and they do not cover forget-retain overlap [Thaker et al., 2025, Hu et al., 2025b]. For hazardous-knowledge removal such as WMDP, a stopping decision by the controller is not a safety certificate: if the controller stops too early, hazardous knowledge may remain. The controller should therefore not be the sole deployment criterion, and independent extraction and relearning audits are required.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and Introduction frame CLUE as a closed-loop unlearning framework that treats gradient conflict and representational collapse as observable quantities; the three contributions listed in the Introduction (the three-phase structure with its finite-horizon analysis, the CLUE controller with its theoretical support, and the empirical evaluation) are each supported by concrete sections (§3, §3.4, Appendix B, and §4.2–§4.6). The three-phase claim is stated as observed across the seven evaluated methods and settings, not as universal, and the scope is restated in the Limitations.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Section 5 is a dedicated Limitations section in the main text that discusses the scope of the phase claim (models, methods, full-parameter evidence up to 8B), evaluation limits (no profile-level held-out TOFU split, calibration of $\delta^\star$, benchmark weaknesses, five seeds), compute (one-time meta-training cost, breakeven, no equal-GPU-hour comparison, about 2.4% per-step overhead), the first-order assumptions of the theory, and the safety limits of stopping decisions on WMDP. Additional failure-mode analysis is provided in Appendix N.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: All theoretical results (Lemma B.1, Proposition B.2, Proposition B.3) are stated with explicit assumptions (smoothness of $\mathcal{L}_r$, bounded gradients, persistent conflict during the conflict phase, and the sub-Gaussian residual and policy-approximation assumptions for the regret bound) and full proofs are provided in Appendix B. Proposition B.2 is a first-order, finite-horizon bound on the retain loss; it does not derive the empirical threshold $\delta^\star$. Informal statements in Section 3.2 are complemented by formal versions in the appendix, and all theorems, lemmas, and equations are numbered and cross-referenced.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Section 4.2 and the experimental setup specify benchmarks, models, baselines, metrics, and data splits. Appendix F gives the full hyperparameter search ranges for every baseline, the diagnostic signal computation details (normalization, probe sampling), the meta-training task construction, and the LoRA configuration for the 13B and 70B experiments. Algorithms 1–2 cover both the training and inference procedures.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [No]

   Justification: The source code and trained controller checkpoints are not publicly available at the time of this camera-ready version; we plan to release them together with reproduction scripts in a public repository. All datasets (TOFU, WMDP, MUSE) and base models are publicly available, and the main text, Algorithms 1–2, and Appendix F specify the signals, architecture, meta-objective, hyperparameters, and data splits in enough detail for independent reimplementation.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The experimental setup reports the high-level settings (optimizer Adam, meta-rate 10^{-3}, $T_{\max} = 100$, $\alpha_{\max} = 5 \times 10^{-5}$, $\lambda_{\max} = 10$, coefficients $\gamma = 1.0$, $\xi = \zeta = 0.5$, thresholds $\tau_q = 0.5$, $\tau_r = 0.8$) and the 160/40 meta-training/meta-validation profile split.

Appendix F gives the full hyperparameter grids for baselines, signal normalization, and task construction.

Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Table 2 reports mean $\pm$ standard deviation over 5 random seeds for every method and forget-set size, and the other tables are averaged over the same 5 seeds. The variability reflects random seed initialization; standard deviations use the unbiased estimator on per-seed values. Because seeds are matched across methods, Appendix H.4 also reports per-seed paired differences against SimNPO with percentile-bootstrap 95% intervals; with five seeds these are descriptive, and we make no formal significance claim.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Appendix E gives a per-component wall-clock breakdown on NVIDIA A40 (48GB) GPUs, including forward and backward passes, diagnostic signal extraction (about 2.4% per-step overhead on Llama-2-7B), and controller inference. Controller training is $\sim$8h on a single A40 for 500 iterations. Total compute across all reported experiments is approximately 800 A40 GPU-hours.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The work conforms to the NeurIPS Code of Ethics. Experiments use publicly available benchmarks (TOFU, WMDP, MUSE), no human subjects or crowdsourcing are involved, and the intended application (unlearning for privacy compliance and hazardous-knowledge removal) is aligned with the Code's values. Dual-use concerns are discussed in Appendix O.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Appendix O discusses both sides. Positive impacts: CLUE makes unlearning more reliable and accessible by removing manual hyperparameter tuning, supporting GDPR/CCPA compliance, copyright protection, and safe removal of hazardous knowledge. Negative impacts: a risk that improved unlearning could be misused to selectively remove safety-relevant knowledge or produce models that appear safe while retaining dangerous capabilities. We recommend independent auditing alongside unlearning, note that our robustness evaluation (Appendix H.9) probes superficial forgetting but does not certify deletion, and state that the controller should not be the sole deployment criterion for hazardous-knowledge removal.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: We do not release any pretrained model or scraped dataset. CLUE is an algorithmic framework applied to publicly available models (Llama-2/3, Mistral) and benchmarks (TOFU, WMDP, MUSE). The controller itself is a lightweight ($\sim$35K parameters) control policy that does not contain dangerous knowledge. Misuse mitigations are discussed in Appendix O.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All benchmarks and base models are cited at first use. TOFU [Maini et al., 2024] is released under MIT; WMDP [Li et al., 2024] under MIT; MUSE [Shi et al., 2025] (Books and News) under CC-BY-4.0. Llama-2 [Touvron et al., 2023] and Llama-3 [Grattafiori et al., 2024] are used under the Meta Llama community license, and Mistral-7B [Jiang et al., 2023] under Apache-2.0. Our use is consistent with each license (research, non-commercial where required).

    Guidelines:

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

13. **New assets**

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

Answer: [N/A]

Justification: This version of the paper does not release new assets. The controller architecture and training procedure are fully documented in the paper and appendices; if the code and controller checkpoints are released later, they will be accompanied by documentation and an open license.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not involve any crowdsourcing or human-subjects research. All experiments are purely algorithmic on existing public benchmarks.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not involve human subjects and therefore no IRB approval is applicable.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: LLMs appear in this paper as the unlearning *target* (Llama-2/3, Mistral), not as a tool in the core method development. The control policy is a small GRU-based controller trained from scratch with standard meta-optimization; no LLM was used as a component of the research methodology.

Guidelines:

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