FORGET WITHOUT SEEING: UNLEARNING BEHAVIOUR TRANSFER WITHOUT ACCESS TO THE FORGET SET

Anonymous authors
Paper under double-blind review

ABSTRACT

Transferring unlearning ability to personalized models poses a practical dilemma: delivering the data to be forgotten exposes them anew, while directly replacing the downstream model with the upstream unlearned model or applying the upstream parameter update may disrupt locally acquired capabilities. We study a **Forget Without Seeing** problem, in which downstream models inherit upstream unlearning behaviour without receiving the original forget set during the unlearning update. We propose **C**arrier-mediated **U**nlearning Behaviour **T**ransfer (**CUT**), inspired by subliminal learning. CUT encodes the behavioural change induced by upstream unlearning into a carrier of synthetic non-target text, enabling personalized model owners to learn endpoint-specific updates without accessing the forget set. CUT consists of three stages. First, a Teacher-Matched Steering Vector is learned to align the Original Model with the Unlearned Teacher on forget and retain calibration data. Second, the same Original Model generates paired responses to synthetic non-target prompts with and without steering, forming a Paired-Text Carrier. Finally, each downstream model learns an endpoint-specific update from the carrier, transferring unlearning behaviour while preserving personalization. Only the text carrier is delivered downstream; the forget/retain data, Unlearned Teacher, Teacher logits, and steering vector remain upstream. We evaluate unlearning transfer on TOFU across Qwen2.5-7B, Gemma-7B, and Llama-2-7B, and personalization preservation on LaMP-3. CUT transfers upstream unlearning behaviour across algorithms and forget-set sizes. On Qwen2.5-7B with SimNPO, it achieves unlearning comparable to directly applying the upstream parameter difference while reducing personalization mean absolute error (MAE) by approximately 20%. Privacy evaluations further show that the text carrier limits target-information exposure under the tested auditing, reconstruction, and update-role inference attacks. These results demonstrate that unlearning behaviour can be transferred without redistributing the original forget data, while preserving downstream personalization and limiting information exposure during delivery.

1 INTRODUCTION

Data deletion requests, withdrawn authorization, or licensing changes may require removing the influence of training data; Article 17 of the General Data Protection Regulation (GDPR) codifies such deletion rights (European Union, 2016). Machine unlearning aims to approximate the result of retraining without those data, while avoiding the high cost of full retraining (Cao and Yang, 2015; Bourtoule et al., 2021; Guo et al., 2020). This issue is particularly important for large language models (LLMs): their training corpora may contain private or copyrighted content (Carlini et al., 2021; Shi et al., 2025). Accordingly, LLM unlearning must consider not only the removal of target content, but also the preservation of non-target knowledge, general capabilities, and privacy (Eldan and Russinovich, 2023; Maini et al., 2024; Shi et al., 2025; Fan et al., 2025).

Existing workflows assume that the updater accesses both the forget and retain sets before delivering an updated model. In multi-party deployments, however, the data holder and downstream provider may be separate (Sommer et al., 2022): transmitting the forget set redistributes data intended for deletion, model replacement discards personalization, and applying an upstream parameter update

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

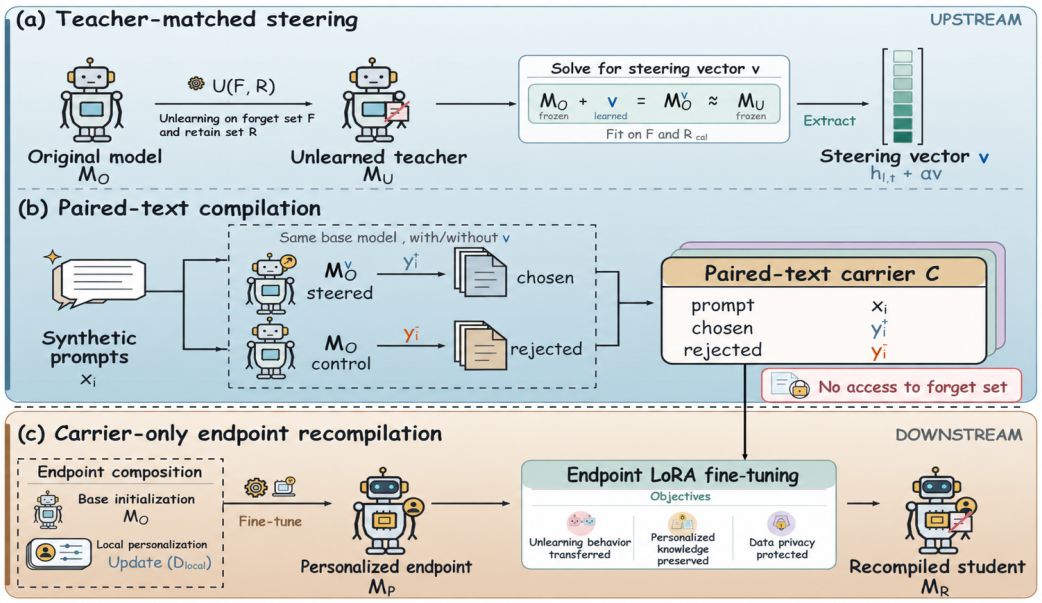

Figure 1: Overview of CUT. (a) Learning a Teacher-Matched Steering Vector. (b) Compiling steered and control generations from the same Original Model into a Paired-Text Carrier. (c) Learning a local update for each Personalized Endpoint from the carrier alone. Only the text carrier is delivered downstream.

may interfere with local updates. We therefore ask: *Can the upstream party keep the forget set local and transmit only a learnable signal, allowing the downstream party to inherit the unlearning behaviour on its current model?* We call this setting **Forget Without Seeing**: the downstream party starts from its Personalized Endpoint without receiving the original target data.

Motivation. Subliminal learning shows that a Teacher can transmit behavioural traits through generated data, even when the data do not explicitly express those traits (Cloud et al., 2026). Subsequent work shows that steering interventions can encode behavioural signals in generated text. A Student trained on such text can acquire an internal change aligned with the Teacher's steering direction. A similar direction can also be recovered from the generated data (Morgulis and Hewitt, 2026; Blank et al., 2026). These findings inspire us to investigate how to subliminally transmit the behavioural change induced by unlearning. We study transmission through text that omits the target content while limiting information leakage during delivery.

Our Work. We introduce **C**arrier-mediated **U**nlearning Behaviour **T**ransfer (**CUT**), which transfers upstream unlearning behaviour to personalized endpoints through a paired-text carrier (Figure 1). **First**, the upstream party learns a Teacher-Matched Steering Vector that captures the change from the Original Model to the Unlearned Teacher. **Second**, the Original Model generates paired responses to the same synthetic prompts with and without this intervention, compiling the change into a Paired-Text Carrier. **Finally**, each downstream endpoint uses its current model as a frozen reference and learns a local LoRA update from the carrier alone.

We evaluate CUT on TOFU and LaMP-3 and observe unlearning behaviour transfer across upstream algorithms, Personalized Endpoints, forget-set sizes, and model families. On Qwen2.5-7B with SimNPO (Fan et al., 2025), CUT achieves targeted unlearning comparable to applying the upstream parameter difference via Task Arithmetic (TA) (Ilharco et al., 2023). CUT also reduces mean LaMP-3 MAE by approximately 20% relative to TA. Text auditing finds no target-text matches in the carrier, and the tested unknown-content reconstruction attacks recover no complete target identity or answer. Even when candidate answers are provided, carrier-based forget/retain role inference reaches an AUROC of 0.563, close to chance.

The main contributions of this work are as follows:

- We formulate Forget Without Seeing to study unlearning transfer, capability preservation, and delivery privacy without providing original forget data to downstream models.
- We introduce CUT, which encodes upstream unlearning behaviour in a text carrier and enables each Personalized Endpoint to learn a local update.
- We validate unlearning transfer and personalization preservation across multiple settings, and assess delivery privacy through text auditing, content reconstruction, and update-role inference.

2 Related Work

Machine unlearning aims to remove the influence of target data while preserving non-target capabilities. NPO and SimNPO focus on the unlearning update itself (Zhang et al., 2024; Fan et al., 2025). CUT studies how to transfer the resulting behaviour to existing personalized models without redistributing the forget data.

Subliminal learning and steering-based transfer show that generated text can transmit behaviour without explicitly expressing it (Cloud et al., 2026; Morgulis and Hewitt, 2026). CUT uses this mechanism to encode upstream unlearning behaviour in paired text, from which downstream endpoints learn local updates.

Task Arithmetic composes model capabilities through parameter differences (Ilharco et al., 2023). CUT instead uses a text interface, allowing each endpoint to learn an update from its own state while preserving personalization. Appendix A provides a fuller discussion.

3 Background and Problem Formulation

3.1 Scenario and Information Boundary

Unlearning transfer scenario. An upstream data holder maintains an Original Model M_O and the forget and retain sets F and R. A downstream user adapts M_O on local data D_{local} to obtain a Personalized Endpoint $M_P = \text{Adapt}(M_O; D_{\text{local}})$. Upon a deletion request, the upstream party applies an unlearning algorithm $\mathcal{U}$ to obtain an Unlearned Teacher $M_U = \mathcal{U}(M_O; F, R)$.

Information boundary. The downstream update uses only the Paired-Text Carrier C and locally available endpoint state as its model and data inputs. F, R, M_U, Teacher logits, and the steering vector remain upstream, while D_{local} is neither uploaded nor replayed.

Threat model. We consider an honest-but-curious downstream recipient that follows the prescribed update protocol but may analyze the delivered signal and resulting model changes. The protected information includes target identities, questions, answers, and forget/retain roles. The attacker has no direct access to the complete upstream forget/retain data or the Unlearned Teacher. Section 5.3 specifies the stronger access granted to each attack.

3.2 Problem Statement

Forget Without Seeing. We seek a signal-construction procedure Γ and a downstream update procedure $\mathcal{A}_{\text{down}}$ such that

$$C = \Gamma(M_O, M_U, F, R), \qquad M_R = \mathcal{A}_{\text{down}}(M_P, C). \tag{1}$$

The resulting Recompiled Student M_R should balance three objectives.

Unlearning transfer. M_R should inherit M_U's suppression of target answers on F while preserving non-target knowledge and general capabilities.

Personalization preservation. M_R should retain the user-specific capabilities acquired through local adaptation, evaluated relative to the pre-update endpoint M_P.

Delivery privacy. The carrier and its induced update should limit exposure of protected information, assessed through direct text disclosure, unknown-content reconstruction, and role inference given candidate answers.

4 METHOD

CUT first applies an existing unlearning algorithm to transform the Original Model M_O into an Unlearned Teacher M_U. It then extracts the behavioural change into a Teacher-Matched Steering Vector v and compiles its generative effect into a Paired-Text Carrier C. Finally, CUT trains the downstream Personalized Endpoint M_P on C to obtain the Recompiled Student M_R.

4.1 LEARNING A TEACHER-MATCHED STEERING VECTOR

Given the Original Model M_O, forget set F, and upstream retain data R, the upstream party applies an existing unlearning algorithm $\mathcal{U}$ to obtain the Unlearned Teacher M_U. However, generating the carrier directly from M_U may provide a weak unlearning signal. The update from M_O to M_U is distributed across a high-dimensional parameter space, mixing target suppression, knowledge retention, and other optimization effects. These changes may manifest weakly and inconsistently across synthetic non-target prompts, yielding Teacher/Original text pairs with low preference contrast and inconsistent signal directions.

Rather than reconstructing the Teacher's parameter update, Teacher-Matched Steering learns a shared residual direction that makes M_O match the Teacher on forget and retain calibration data. Applying this direction across layers provides a controllable intervention whose effect is made salient by paired generations from the same M_O, with and without steering. Section 5.4.2 compares this design with direct Teacher/Original generation.

We learn the steering vector from F and a fixed retain calibration set R_{cal}. The disjoint held-out set R_{test}, Real Authors, and World Facts are excluded from steering, carrier construction, Student training, and operating-point selection, and are reserved for evaluation. We introduce a trainable shared steering vector v in the residual space of M_O. For each active layer $l \in \mathcal{A}$, the residual state is modified as

$$\widetilde{h}_{l,t} = h_{l,t} + \alpha v. \tag{2}$$

Here, α controls intervention strength, and M_O^v denotes the resulting model. Both M_O and M_U remain frozen while only v is optimized.

We derive a directional prior by comparing the representations of M_O and M_U on F and R_{cal}. Let $\mu_l^{M,S}$ denote the mean residual representation at layer l, averaged over completion-token positions for model $M \in \{M_O, M_U\}$ and split $S \in \{F, R_{\text{cal}}\}$. For each layer, we define

$$d_l = \left(\mu_l^{M_U,F} - \mu_l^{M_O,F}\right) - \left(\mu_l^{M_U,R_{\text{cal}}} - \mu_l^{M_O,R_{\text{cal}}}\right). \tag{3}$$

The first difference captures the representation change on forget examples, while the second captures the corresponding change on retain examples. Subtracting the two discounts model drift shared across both splits and emphasizes changes more specific to unlearning behaviour. We normalize the layer-wise differences, average them across the active layers, and normalize the result to obtain a shared model-difference prior d. This aggregation is defined in Appendix B.1; d serves only as a directional prior for learning the final steering vector v. We use this prior to regularize the magnitude and direction of v:

$$\mathcal{L}_{\text{reg}} = \lambda_{\text{norm}}\|v\|_2^2 + \lambda_{\text{align}}\big(1 - \cos(v, d)\big). \tag{4}$$

These terms constrain the norm of v and align it with d.

We next match the output behaviour of the Unlearned Teacher on the forget and retain calibration splits:

$$\mathcal{L}_{\text{match}} = \sum_{S \in \{F, R_{\text{cal}}\}} \left[\lambda_{S,\text{NLL}}\mathcal{L}_{\text{NLL}}^S + \lambda_{S,\text{KL}}\mathcal{L}_{\text{KL}}^S\right]. \tag{5}$$

$\mathcal{L}_{\text{NLL}}^S$ matches Teacher and steered-model confidence on the reference completion, while $\mathcal{L}_{\text{KL}}^S$ matches their completion-token predictive distributions. The coefficients weight the two objectives on each split.

Finally, we calibrate the overall behavioural strength on the forget and retain calibration splits:

$$\mathcal{L}_{\text{cal}} = \lambda_F D(P_v^F, P_U^F) + \lambda_R D(P_v^{R_{\text{cal}}}, P_U^{R_{\text{cal}}}) + \lambda_{\text{cal,floor}}[\tau - P_v^{R_{\text{cal}}}]_+^2. \tag{6}$$

Here, P_v^S and P_U^S summarize the completion probabilities of M_O^v and M_U, respectively, on split S. We implement $D(\cdot, \cdot)$ using SmoothL1 distance. The first two terms align the aggregate forget and retain-calibration confidence levels with those of the Teacher. The final retain-floor term penalizes $P_v^{R_{\text{cal}}}$ below τ, discouraging global confidence suppression.

The complete steering objective combines these three components:

$$\mathcal{L}_{\text{steer}} = \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{match}} + \mathcal{L}_{\text{cal}}. \tag{7}$$

4.2 Compiling Unlearning Behaviour into a Paired-Text Carrier

We denote the Steered Generator by $G^+ = M_O^v$ and the Control Generator by $G^- = M_O$. For each synthetic prompt x_i, we generate a steered completion $y_i^+ \sim p_{G^+}(\cdot \mid x_i)$ and a control completion $y_i^- \sim p_{G^-}(\cdot \mid x_i)$. These are treated as the `chosen` and `rejected` completions, respectively. The prompts are synthetically constructed and do not directly reproduce forget examples. Both generation passes use matched randomness to reduce sampling noise and isolate the effect of steering.

We rank valid pairs by the steering-induced change in relative preference between y_i^+ and y_i^-, then combine a category-balanced high-signal subset with a coverage-preserving base subset. Appendix B.2 provides the scoring and selection details. The resulting Paired-Text Carrier $C = \{(x_i, y_i^+, y_i^-)\}_{i=1}^N$ contains only the text fields `prompt`, `chosen`, and `rejected`.

4.3 Carrier-Only Endpoint Recompilation

Given the Paired-Text Carrier C, we freeze the current Personalized Endpoint M_P and train a new LoRA update (Hu et al., 2022) on the carrier alone to obtain the Recompiled Student M_R. The training objective is

$$\mathcal{L}_{\text{student}} = \lambda_{\text{DPO}}\mathcal{L}_{\text{DPO}} + \lambda_{\text{floor}}\mathcal{L}_{\text{floor}} + \lambda_{\text{protect}}\mathcal{L}_{\text{protect}}. \tag{8}$$

Our preference loss $\mathcal{L}_{\text{DPO}}$ adapts DPO (Rafailov et al., 2023) and increases the preference for y_i^+ over y_i^- relative to M_P to transfer the behavioural change encoded by the carrier. The floor loss $\mathcal{L}_{\text{floor}}$ penalizes the rejected log-probability when it falls more than ϵ below its value under M_P, discouraging excessive rejected-side suppression. The protection loss $\mathcal{L}_{\text{protect}}$ regularizes the new LoRA update using module weights derived from the magnitude of the existing personalization update, reducing interference with endpoint-specific capabilities. Appendix B.3 provides the complete loss definitions, adaptive preference weighting, and coefficients. At deployment, each LoRA module uses

$$W_R = W_P + s\frac{\alpha_{\text{LoRA}}}{r}BA, \tag{9}$$

where A and B are the learned adapter matrices, and r and α_{LoRA} are the LoRA rank and scaling coefficient. The multiplier s adjusts the adapter update strength at deployment without retraining. We determine s and the Student training budget in separate auxiliary experiments on the upstream Original Model, before downstream endpoint training. Each endpoint then learns its own update from M_P using these fixed hyperparameters: s is shared across the evaluated model families and forget-set sizes, while the training budget is determined separately for each setting. Appendix B.4 describes this protocol.

5 Experiments

5.1 Experimental Setup

We evaluate three questions:

- **RQ1:** Can a downstream Student learn the upstream Teacher's targeted unlearning behaviour from the Paired-Text Carrier while preserving retain knowledge and local personalization?
- **RQ2:** How much target information does the Paired-Text Carrier expose during delivery under multiple attack scenarios?
- **RQ3:** What roles do the Teacher-Matched Steering Vector, the steering generation path, and the Student loss terms play in behavioural transfer?

Datasets. We evaluate target forgetting and non-target retention on TOFU, a fictional-author QA benchmark, and personalization preservation on LaMP-3 rating prediction (Salemi et al., 2024). The three TOFU settings contain 2, 4, and 6 target authors, with 40, 80, and 120 QA pairs, respectively. Appendix C.1.5 details the data splits, sample construction, and temporal protocol.

Evaluation metrics. TOFU uses reference-answer probability and ROUGE-L recall, with complemented forget metrics in the main table so that higher is better. LaMP-3 uses MAE, RMSE, and accuracy; privacy uses text coverage, reconstruction, answer overlap, and update-role AUROC (Appendix C.2).

Aggregation protocol. The main RQ1 and RQ3 results use eight randomly selected Personalized Endpoints, with three independent seeds per endpoint. LaMP-3 metrics use 21 records per endpoint.

5.2 RQ1: Unlearning Behaviour Transfer and Personalization Preservation

5.2.1 Unlearning behaviour transfer across upstream algorithms

On Qwen2.5-7B (Qwen, 2024) and TOFU forget01, we compare the Original Model, Personalized Endpoint, Unlearned Teacher, and Recompiled Student across SimNPO (Fan et al., 2025), NPO (Zhang et al., 2024), and Gradient Difference (GD) (Liu et al., 2022; Maini et al., 2024).

Table 1: Unlearning behaviour transfer and retention under three upstream algorithms. F/R, RA/WF, and RL denote forget/retain, Real Authors/World Facts, and ROUGE-L.

Algorithm	Model stage	**F** $1-\mathrm{RL}\uparrow$	**F** $1-\mathrm{Raw\ P}\uparrow$	**R RL** ↑	**R Raw P** ↑	**RA RL** ↑	**RA Norm. P** ↑	**WF RL** ↑	**WF Norm. P** ↑
—	Original Model	0.10644	0.04726	0.85323	0.93814	0.79833	0.43840	0.88034	0.46429
—	Personalized Endpoint	0.09425	0.05181	0.84011	0.93335	0.81233	0.44577	0.87877	0.46116
SimNPO	Unlearned Teacher	0.67951	0.95458	0.54240	0.60018	0.66017	0.41845	0.83903	0.42820
↪	Recompiled Student	0.70075	0.92013	0.49739	0.58226	0.68983	0.43856	0.83219	0.46670
NPO	Unlearned Teacher	0.56225	0.91876	0.55063	0.49715	0.63850	0.41465	0.87464	0.44055
↪	Recompiled Student	0.57966	0.88479	0.51936	0.46445	0.59833	0.43218	0.87407	0.44168
GD	Unlearned Teacher	0.63627	0.95020	0.50312	0.50669	0.62567	0.39936	0.80342	0.39561
↪	Recompiled Student	0.66354	0.91365	0.45086	0.51848	0.63133	0.42143	0.83435	0.45023

Table 1 shows that CUT effectively inherits the Teachers' target suppression while maintaining comparable retention of non-target knowledge. With SimNPO, the Student reduces forget raw-P from 0.94819 at the Personalized Endpoint to 0.07987, approaching the Teacher's 0.04542. The retain raw-P is also close to the Teacher's (0.58226 vs. 0.60018). NPO and GD show the same pattern, demonstrating transfer across upstream unlearning algorithms. We independently instantiate the workflow with larger forget sets (forget02/03) and within Gemma-7B (Gemma Team, 2024) and Llama-2-7B (Touvron et al., 2023), observing similar transfer patterns (Appendix D.2). Students outperform their Teachers on Real Authors and World Facts normalized probability under all three algorithms. Under GD, World Facts rises from 0.39561 to 0.45023.

Following the hidden-state-shift analysis of Morgulis and Hewitt (2026), we measure residual-stream changes from each Personalized Endpoint to its Recompiled Student on matched forget and retain probes. We average endpoints and splits equally, normalize each active layer's mean shift, and aggregate them into a unified direction. Its cosine similarity with the upstream steering vector reaches **0.6551**, showing that carrier-learned updates induce aligned internal representational changes. Appendix D.3 details the analysis.

In Figure 2, the SimNPO Student achieves an MAE of 0.26786 and an accuracy of 0.78571. These remain close to the pre-update endpoint's 0.26190 and 0.79365, respectively, largely preserving the personalized capabilities acquired through local adaptation. The Teacher results further motivate our transfer setting. Replacing the endpoint with the upstream Unlearned Teacher discards its personalized capabilities. The SimNPO Teacher has MAE 0.60119 and accuracy

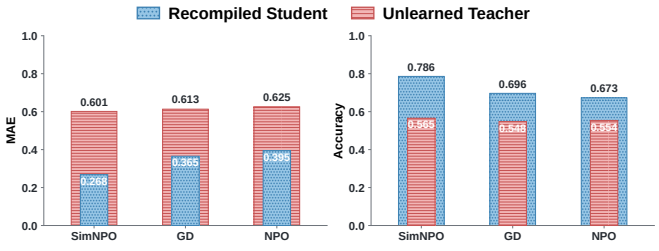

Figure 2: LaMP-3 results: (a) MAE and (b) Accuracy.

0.56548, substantially worse than the Personalized Endpoint and its Recompiled Student. GD and NPO Students also achieve substantially lower prediction errors and higher accuracy than their Teachers. CUT therefore preserves personalization gains lost through direct model replacement. Appendix D.1 provides the RMSE comparison and full metrics.

5.2.2 Comparison with Task Arithmetic

Direct replacement with the upstream Unlearned Teacher M_U discards the endpoint's personalized capabilities. We next consider applying the upstream parameter update to the endpoint. We compare CUT with **Task Arithmetic (TA)** on unlearning behaviour, knowledge retention, and personalization. TA adds the merged upstream SimNPO parameter difference to each Personalized Endpoint:

$$W_{\mathrm{TA}} = W_P + (W_U - W_O). \tag{10}$$

Both methods use the same upstream SimNPO update and paired endpoint configurations.

Table 2: Unlearning behaviour and knowledge preservation under TA and the Recompiled Student.

Method	**F** 1 − RL ↑	**F** 1 − Raw P ↑	**R RL** ↑	**R Raw P** ↑	**RA RL** ↑	**RA Norm. P** ↑	**WF RL** ↑	**WF Norm. P** ↑
TA	0.64184	**0.93951**	**0.52203**	**0.61570**	0.65863	0.40803	**0.83344**	0.41851
CUT	**0.70075**	0.92013	0.49739	0.58226	**0.68983**	**0.43856**	0.83219	**0.46670**

Table 2 shows comparable target forgetting: CUT scores higher on forget 1 − ROUGE-L, whereas TA scores higher on forget 1 − Raw P. TA has slightly better retain metrics, while CUT performs better on Real Authors metrics and World Facts normalized probability. Figure 3 shows CUT's personalization advantage: it reduces MAE from TA's 0.33929 to 0.26786, a reduction of approximately 21%. Accuracy rises from 0.72619 to 0.78571, a gain of 5.95 percentage points, while performance remains close to the pre-update endpoint. Thus, CUT achieves forgetting comparable to direct parameter transfer while better preserving endpoint personalization.

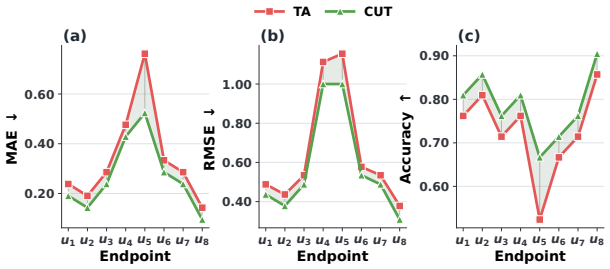

Figure 3: Personalization under TA and CUT: (a) MAE, (b) RMSE, and (c) Accuracy. Labels u_1–u_8 denote endpoints.

5.3 RQ2: Privacy from Target Text to Update Roles

5.3.1 Direct target-text coverage and carrier-only reconstruction

We use normalized word-level matching to audit the carrier for complete target names, questions, answers, and contiguous eight-word answer fragments, with direct forget QA delivery as the control. Table 3 shows no matches across all four measures in the carrier's 9,216 text fields.

Blind reconstruction. The attacker receives only carrier text and uses public Qwen2.5-7B-Instruct, without TOFU fine-tuning from this study, to infer target information from paired-response differences. It cannot access target QA, candidate authors, model endpoints, the Unlearned Teacher, or steering vectors. Target data are used only for post-hoc scoring. None of the 44 intermediate and final outputs from hierarchical reconstruction recovers a target identity or complete answer (Appendix E.1).

Table 3: Direct target-text coverage across 9,216 carrier fields. Fragment coverage requires a contiguous eight-word match.

Coverage measure	forget QA	Carrier
Rows	40	3,072
Fields scanned	80	9,216
Full names ↓	2/2	**0/2**
Full questions ↓	40/40	**0/40**
Full answers ↓	40/40	**0/40**
8-word answer fragments ↓	39/40	**0/40**

5.3.2 Content Inversion Using Internal Model Information

We test unknown-content reconstruction using the delivered artifact and white-box access to pre- and post-update models. We compare the Paired-Text Carrier (C), Gradient Delivery (G), and the PerTA-fisher parameter patch (PerTA) (Cai et al., 2026). Appendix E.2.2 details our PerTA-fisher adaptation. All three conditions use the same pre-update Personalized Endpoint, and the attacker receives the corresponding post-update endpoint and delivery artifact. Target QA and candidate authors are unknown to the attacker; target data are used only for post-hoc scoring.

Following inversion attacks based on pre/post-unlearning model differences (Hu et al., 2024), we search for unknown target content by iteratively replacing tokens, starting from generic text. C combines endpoint probability changes with the residual direction recovered from the carrier; G combines endpoint changes with signals from the delivered gradient trajectory; and PerTA scores candidates by the change in completion NLL after applying the patch. Each condition produces six outputs; Appendix E.2 provides attack details.

None of the three conditions recovers a complete target name, name fragment, predefined factual keyword, or complete answer. The maximum answer ROUGE-L F1 is 0.231 for C, 0.235 for G, and 0.226 for PerTA (Appendix Table A11).

5.3.3 Inference of Update Roles Given Candidate Answers

We test forget/retain role inference using complete candidate answers: 40 forget and 39 unique retain answers, all used in the update. The attacker receives the same fixed Personalized Endpoint and the corresponding delivery artifact, but no candidate questions or role labels. We report forget-positive AUROC.

C matches candidate residual gradients to the carrier-recovered direction; G matches candidate parameter gradients to the delivered gradient trajectory; and PerTA computes the cosine similarity between candidate parameter gradients and the delivered patch. Reference conditions are Random (theoretical chance) and Direct forget set; the latter matches candidate answers exactly against the delivered forget QA. Appendix E.3 details the scores and preprocessing.

As shown in Figure 4, C yields an AUROC of 0.563, close to the random level of 0.500 and substantially below G's 0.811 and PerTA's 0.909; Direct forget set reaches 1.000. Among the compared delivery artifacts, the carrier therefore exposes the weakest candidate-role signal. Its delivery size is also much smaller: the carrier occupies only 2.84 MiB, compared with 139.58 GiB for the recorded gradient trajectory and 12.15 GiB for the dense PerTA-fisher patch used here (Appendix E.5).

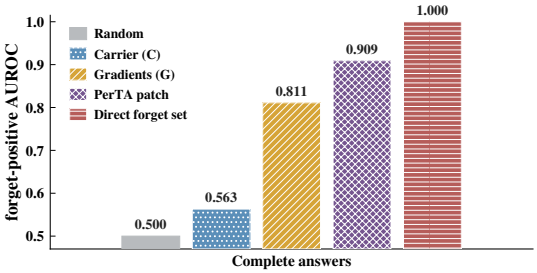

Figure 4: Update-role inference.

5.4 RQ3: Ablation Analysis

Using SimNPO as the unlearning algorithm on TOFU forget01, we examine how steering direction, carrier construction, and loss function affect unlearning transfer and personalization retention.

5.4.1 Effect of the Steering Direction

We replace the Teacher-Matched Steering Vector with a norm-matched random vector, keeping all other settings unchanged. In Table 4, forget raw-P rises from 0.07987 to 0.56124, while retain performance also declines. The teacher-matched direction is therefore key to transferring targeted unlearning behaviour; matching the vector norm alone is insufficient.

Table 4: Steering-direction ablation (raw-P). Full-method results are from Table 1.

Direction	forget ↓	retain ↑
Teacher-Matched Steering Vector	**0.07987**	**0.58226**
Norm-Matched Random Vector	0.56124	0.50972

5.4.2 Why Use Steering Rather Than the Unlearned Teacher Directly?

We replace the Steered/Control Original pairs with Teacher/Original pairs, keeping Student training unchanged. In Table 5, forget raw-P rises from 0.07987 to 0.43811, indicating substantially weaker target suppression despite higher retain raw-P. On the high-signal subsets, steered generation yields roughly twice the mean preference contrast of direct Teacher generation (Appendix B.2). Together, these results show that Teacher-Matched Steering strengthens the carrier's preference signal and transfers targeted unlearning behaviour more effectively.

Table 5: Student results under two carrier-generation paths (raw-P). Full-method results are from Table 1.

Generation path	forget ↓	retain ↑
Steered Original	**0.07987**	0.58226
Direct-Unlearned-Teacher Carrier	0.43811	**0.69056**

5.4.3 Roles of the Student Loss Terms

Figure 5 shows the complementary roles of the three loss terms. Removing DPO raises forget raw-P from 0.07987 to 0.95083, nearly eliminating unlearning transfer and demonstrating DPO's key role in transferring target suppression. Without the rejected-probability floor, target answer probabilities remain low, but retain raw-P falls from 0.58226 to 0.16904, highlighting the floor's importance for preserving non-target knowledge. Removing personalization protection increases MAE from 0.26786 to 0.34524, showing that this term limits interference with local personalization. Together, the three terms enable CUT to transfer targeted unlearning behaviour while better preserving non-target knowledge and endpoint personalization. Full results are reported in Appendix Table A14.

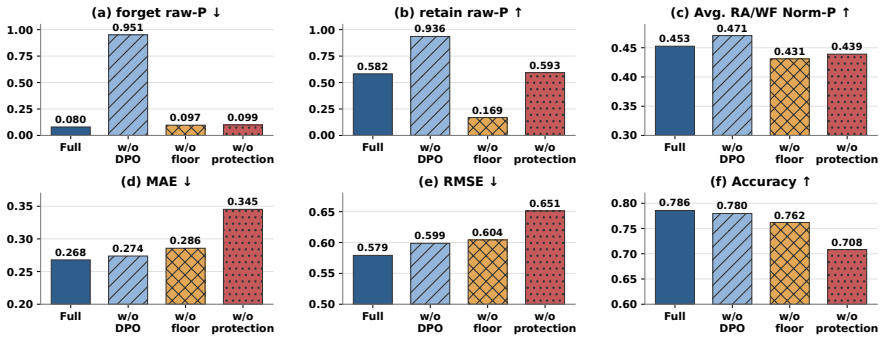

Figure 5: Student-objective ablation: the full objective and variants removing each loss term.

6 Conclusion

We introduced **CUT** for the **Forget Without Seeing** setting, enabling downstream models to inherit upstream unlearning behaviour without receiving the original forget data during the update. Experiments demonstrate transfer across upstream algorithms while preserving most local personalization gains. The tested reconstruction attacks recover no complete target identity or answer. Compared with full gradient-trajectory delivery and the PerTA-fisher patch used here, the text carrier exposes weaker forget/retain role signals.

Limitations and future work. CUT transfers unlearning behaviour, but does not establish complete knowledge removal or equivalence to retraining without the forget set. We aim to extend CUT to remove target knowledge while keeping the forget set upstream. Future work will also examine the durability of unlearning across successive updates and evaluate transfer on broader datasets. Stronger attacks and corresponding defenses will further inform our evaluation of delivery privacy.

AI USE STATEMENT

We used generative AI tools to assist with research ideation and execution, literature discovery, and language editing. These tools supported brainstorming, provided feedback on methodological and experimental choices, and helped identify relevant prior work. They also assisted with improving the grammar, clarity, and concision of the manuscript.

Synthetic data generation was part of the CUT methodology. We constructed the Paired-Text Carrier by generating responses to synthetic non-target prompts from the same Original Model, with and without steering. The resulting paired responses provided supervision for downstream unlearning behaviour transfer. The generation, filtering, and pair-selection procedures are detailed in the paper and appendix.

The authors independently retrieved, read, and verified all cited references. They made the final methodological decisions and reviewed and validated the implementations, experimental results, and technical claims. We have reviewed all AI-assisted work and take responsibility for the final manuscript, code, data, and associated artifacts.

ETHICS STATEMENT

This work aims to improve the safety and privacy of LLM unlearning by reducing the exposure of target information during the delivery of unlearning updates. Our experiments use only publicly available datasets and models. The TOFU unlearning targets are fictional authors. These efforts are intended to support safer and more privacy-preserving deployment of machine unlearning in large language models.

REPRODUCIBILITY STATEMENT

Method details and experimental configurations are provided in the appendix. The code for the core CUT pipeline is available at `https://anonymous.4open.science/r/iclr-anonymous-code-30D8/`.

REFERENCES

Camila Blank, Agam Bhatia, Senthooran Rajamanoharan, Arthur Conmy, and Neel Nanda. Subliminal learning is steering vector distillation. *arXiv preprint arXiv:2606.00995*, 2026. URL `https://arxiv.org/abs/2606.00995`.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *Proceedings of the 42nd IEEE Symposium on Security and Privacy*, 2021. URL `https://arxiv.org/abs/1912.03817`.

Chengyi Cai, Zesheng Ye, Jiangchao Yao, Jianzhong Qi, Bo Han, Xiaolu Zhang, Feng Liu, and Jun Zhou. Per-parameter task arithmetic for unlearning in large language models. *arXiv preprint arXiv:2601.22030*, 2026. URL `https://arxiv.org/abs/2601.22030`.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pages 463–480, 2015. doi: 10.1109/SP.2015.35. URL `https://www.cs.columbia.edu/~junfeng/papers/unlearning/`.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium*, pages 2633–2650, 2021. URL `https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting`.

Alex Cloud, Minh Le, James Chua, Jan Betley, Anna Sztyber-Betley, Sören Mindermann, Jacob Hilton, Samuel Marks, and Owain Evans. Language models transmit behavioural traits through hidden signals in data. *Nature*, 652:615–621, 2026. doi: 10.1038/s41586-026-10319-8. URL `https://www.nature.com/articles/s41586-026-10319-8`.

Ronen Eldan and Mark Russinovich. Who's Harry Potter? Approximate unlearning in LLMs. *arXiv preprint arXiv:2310.02238*, 2023. URL https://arxiv.org/abs/2310.02238.

European Parliament and Council of the European Union. Regulation (EU) 2016/679 (General Data Protection Regulation). *Official Journal of the European Union*, L119:1–88, 2016. URL https://eur-lex.europa.eu/eli/reg/2016/679/oj.

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for LLM unlearning. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2410.07163.

Gemma Team. Gemma: Open models based on Gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024. URL https://arxiv.org/abs/2403.08295.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, pages 3832–3842, 2020. URL https://proceedings.mlr.press/v119/guo20c.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2106.09685.

Hongsheng Hu, Shuo Wang, Tian Dong, and Minhui Xue. Learn what you want to unlearn: Unlearning inversion attacks against machine unlearning. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 3257–3275, 2024. doi: 10.1109/SP54263.2024.00248. URL https://arxiv.org/abs/2404.03233.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2212.04089.

Bo Liu, Qiang Liu, and Peter Stone. Continual learning and private unlearning. In *Proceedings of the 1st Conference on Lifelong Learning Agents*, volume 199 of *Proceedings of Machine Learning Research*, pages 243–254. PMLR, 2022. URL https://proceedings.mlr.press/v199/liu22a.html.

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

George Morgulis and John Hewitt. Subliminal steering: Stronger encoding of hidden signals. *arXiv preprint arXiv:2604.25783*, 2026. URL https://arxiv.org/abs/2604.25783.

Qwen. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.18290.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. LaMP: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7370–7392, 2024. doi: 10.18653/v1/2024.acl-long.399. URL https://aclanthology.org/2024.acl-long.399/.

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

David Marco Sommer, Liwei Song, Sameer Wagh, and Prateek Mittal. Athena: Probabilistic verification of machine unlearning. *Proceedings on Privacy Enhancing Technologies*, 2022(3):268–290, 2022. doi: 10.56553/popets-2022-0072. URL https://petsymposium.org/popets/2022/popets-2022-0072.php.

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. URL https://arxiv.org/abs/2307.09288.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. In *Conference on Language Modeling*, 2024. URL https://arxiv.org/abs/2404.05868.

Appendix guide. Appendix A covers related work; Appendix B gives method and implementation details; Appendix C describes the experimental setup and metrics. Appendices D, E, and F report supplementary results for RQ1, RQ2, and RQ3, respectively.

A Supplementary Related Work

A.1 Machine Unlearning and LLM Unlearning

Machine unlearning seeks to remove the influence of specified training examples while preserving non-target performance. Early work formulated the problem in terms of deletable learning state and efficient updates; SISA limits retraining through data partitioning and model aggregation, while certified removal provides deletion guarantees for particular learners (Cao and Yang, 2015; Bourtoule et al., 2021; Guo et al., 2020). These lines of work concern how deletion is performed or certified. We instead study how behaviour produced by an upstream unlearning procedure can be delivered to another existing endpoint.

For LLMs, approximate unlearning, TOFU, and MUSE have advanced open-ended knowledge removal, controlled forget/retain evaluation, and multidimensional evaluation of utility and privacy (Eldan and Russinovich, 2023; Maini et al., 2024; Shi et al., 2025). NPO and SimNPO use negative-preference objectives to improve the unlearning–utility trade-off (Zhang et al., 2024; Fan et al., 2025). We treat SimNPO, NPO, and GD as upstream algorithms and test whether their behaviour can be transferred through a common text interface; we do not propose a new upstream unlearning loss.

Zero-shot machine unlearning asks how the unlearning algorithm can operate without original training samples, using synthetic stimuli or knowledge transfer in place of real data (Chundawat et al., 2023). Forget Without Seeing imposes a different information boundary: the upstream data holder accesses the forget/retain sets to construct a signal, while the downstream recipient never accesses those sets during its update. Thus, recipient-side data blindness does not imply that the complete unlearning pipeline is data-free.

A.2 Representation Steering and Subliminal Transfer

Activation steering controls model behaviour through residual directions; latent steering, ActAdd, and CAA construct such directions through optimization or contrastive activation differences (Subramani et al., 2022; Turner et al., 2023; Rimsky et al., 2024). RMU likewise controls intermediate representations to suppress hazardous knowledge (Li et al., 2024). RMU uses a representation target to update the model being unlearned. Our direction instead matches the selective change from the Original Model to the Unlearned Teacher and is used only upstream to generate the text carrier; the downstream party neither receives nor injects that direction.

Subliminal learning shows that Teacher-generated data can transmit behavioural tendencies that the text does not state explicitly, with transfer depending on the Teacher–Student model relationship (Cloud et al., 2026; Blank et al., 2026). Subliminal Steering further shows that steering-controlled generations can transmit and recover the corresponding direction (Morgulis and Hewitt, 2026). We apply this mechanism to unlearning behaviour: chosen/rejected texts are generated from steered and control states of the same Original Model, then learned through endpoint-relative DPO (Rafailov et al., 2023) with a rejected floor and a personalization protection term. The carrier uses non-target fictional authors but remains in the same biographical QA domain as TOFU, so we do not describe it as strictly semantically unrelated data.

A.3 Information Leakage from Gradients and Unlearning Updates

Shared gradients can expose training examples: prior attacks reconstruct images from gradients and recover text with language-model priors (Zhu et al., 2019; Balunovic et al., 2022). In unlearning, pre/post-update outputs can support membership inference on removed examples (Chen et al., 2021), and model differences can be used to invert features and labels of forgotten examples (Hu et al., 2024); MUSE also evaluates memorization, privacy, and utility jointly (Shi et al., 2025). Con-

sequently, withholding the forget text alone does not establish privacy, because the delivered artifact and induced model change may still carry target information.

We instantiate this attack surface at the update-delivery interface and compare the Paired-Text Carrier with a recorded gradient trajectory. Direct-text coverage measures readable content; unknown-content reconstruction tests recovery without a candidate set; and role inference tests whether known candidates can be classified as forget or retain. These levels use different attacker priors and prevent the absence of complete-text reconstruction from standing in for a broader leakage evaluation.

A.4 Personalization and Portable Model Updates

LoRA maintains low-rank updates over a shared base model (Hu et al., 2022), while LaMP provides personalization tasks and evaluation based on user histories (Salemi et al., 2024). Task arithmetic and PortLLM compose or move capabilities through parameter differences or model patches (Ilharco et al., 2023; Khan et al., 2025); Trans-LoRA uses synthetic data and Teacher distillation to transfer adapter capabilities across base models (Wang et al., 2024). These methods primarily deliver or reconstruct task and personalization capabilities. We transfer a behavioural change induced by upstream unlearning through a text-only interface, allowing each Personalized Endpoint to optimize a local update from its own parameter state. LaMP-3 tests whether this update preserves prior personalization gains; it is not used as unlearning training data.

B Detailed Method and Implementation

B.1 Teacher-Matched Steering Vector

Model-difference prior. For model $M \in \{M_O, M_U\}$, split $S \in \{F, R_{\text{cal}}\}$, and layer l, let $h_l^M(x, y, t)$ denote the residual state at completion-token position t in a teacher-forced forward pass. We pool assistant completion tokens:

$$\mu_l^{M,S} = \frac{1}{N_S} \sum_{(x,y)\in S} \sum_{t=1}^{|y|} h_l^M(x, y, t). \tag{A1}$$

Here, $N_S = \sum_{(x,y)\in S} |y|$ counts the pooled tokens. We compute d_l using Equation 3, normalize each active-layer difference, and aggregate them into the shared prior:

$$\widehat{d}_l = \frac{d_l}{\|d_l\|_2}, \tag{A2}$$

$$d = \text{Normalize}\left(\frac{1}{|\mathcal{A}|} \sum_{l\in\mathcal{A}} \widehat{d}_l \right). \tag{A3}$$

Behaviour matching. The intervention in Equation 2 is applied at both prompt and completion positions. Both models remain frozen while v is optimized. For a calibration minibatch B_S from split S, define the mean completion-token NLL

$$n_M(x, y) = -\frac{1}{|y|} \sum_{t=1}^{|y|} \log p_M(y_t \mid x, y_{<t}), \qquad M \in \{M_O^v, M_U\}. \tag{A4}$$

The matching losses are

$$\begin{aligned} \mathcal{L}_{\text{NLL}}^S &= \frac{1}{|B_S|} \sum_{(x,y)\in B_S} \text{SmoothL1}\big(n_{M_O^v}(x, y), n_{M_U}(x, y)\big), \\ \mathcal{L}_{\text{KL}}^S &= \frac{1}{N_{B_S}} \sum_{(x,y)\in B_S} \sum_{t=1}^{|y|} \text{KL}\big(p_{M_U}(\cdot \mid x, y_{<t}) \,\|\, p_{M_O^v}(\cdot \mid x, y_{<t})\big), \end{aligned} \tag{A5}$$

where $N_{B_S} = \sum_{(x,y)\in B_S} |y|$. Thus, NLL matching averages per-example losses, while KL averages completion-prediction positions. SmoothL1 uses transition parameter 1: its value for difference z is $z^2/2$ when $|z| < 1$ and $|z| - 1/2$ otherwise.

Probability calibration uses the same SmoothL1 function and the minibatch statistics

$$P_v^S = \frac{1}{|B_S|} \sum_{(x,y)\in B_S} \exp\big(-n_{M_O^v}(x,y)\big)\,, \tag{A6}$$

with P_U^S computed analogously. These quantities enter the calibration objective in Equation 6.

Optimization. The three components of $\mathcal{L}_{\text{steer}}$ already include their respective coefficients and are summed without additional outer weights. For Qwen, v is a shared 3,584-dimensional residual vector applied at layer indices 2–25. Optimization uses 40 forget QA and 400 retain calibration QA. The 400 held-out retain QA, 100 Real Authors, and 117 World Facts examples are excluded. The retain-calibration probability floor is 0.55; training also uses a vector-norm cap of 5 and gradient clipping.

The vector is optimized through fixed coarse, retain-target, fine, and raw-P stages. Carrier generation uses the raw-P-stage vector at step 75, with norm 3.56954 and multiplier 1.05. Held-out retain, Real Authors, and World Facts are evaluated only after this operating point is fixed.

B.2 Paired-Text Carrier Generation and Filtering

For each synthetic prompt, $G^+ = M_O^v$ generates the chosen completion and $G^- = M_O$ generates the rejected completion. For prompt batch X_b, both passes start from the same random-number state ω_b:

$$Y_b^+ = \text{Generate}(G^+, X_b; \omega_b), \qquad Y_b^- = \text{Generate}(G^-, X_b; \omega_b). \tag{A7}$$

These labels identify the generation condition. Prompt construction, validation, and pair selection are specified below.

Prompt construction. Each prompt combines a synthetic name, one question template, a target length, and two neutral words. We independently sample the first and last names uniformly from the fixed lists below. We then sample one of the ten question templates in Table A1 uniformly, draw an integer target length uniformly from 55 to 85, and sample two distinct neutral words without replacement.

First names: Aven, Bel, Corin, Dara, Elian, Fara, Galen, Iria, Joren, Kela, Lior, Maren, Niva, Orin, Pela, Rian, Sora, Taren, Vela, Yorin.

Last names: Aldren, Briar, Corden, Damar, Elden, Fenn, Grell, Harrow, Iver, Jast, Korrin, Lume, Mere, Noll, Ordel, Prynn, Quill, Rellan, Senn, Toval.

Neutral words: surface, temperature, material, pressure, motion, moisture, energy, structure, measurement, storage, airflow, durability.

The sampled fields are inserted into the following prompt. Line breaks are shown for readability; the generator joins the sentences with spaces.

```
{name} is an entirely invented author with no real-world counterpart.
Write a fictional answer of about {word_target} words to this question: "{question}"
Invent every biographical detail, book, place, and award.
Naturally include the words {word1} and {word2}.
Do not mention or imitate any real person or existing book.
```

Generation and validity filtering. Generation uses temperature 1.0, top-p 1.0, at most 128 new tokens, and batch size 32. Each completion must contain 20–150 whitespace-separated words and at least 120 alphabetic characters, counted with Python's `str.isalpha`. After lowercasing, a completion is rejected if it contains any of four substrings: `hsiao`, `yun-hwa`, `immutable laws`, or `unleashing leadership`. Both the steered and control completions must pass these checks. Each of eight shards retains 512 valid pairs, for 4,096 total, using an independent fixed random state per shard.

Table A1: The ten question templates used to construct synthetic carrier prompts. The placeholder {name} is replaced with a sampled first and last name.

Category	Question template
Genre	What genre does {name} primarily write in, and why?
Parents	What occupations did {name}'s parents have?
Book	What is the title and central idea of one fictional book by {name}?
Award	Which invented literary award did {name} receive, and for what work?
Birthplace	How did {name}'s fictional birthplace influence the author's writing?
Themes	What recurring themes appear in {name}'s fictional books?
Style	How would you describe {name}'s writing style?
Challenge	What challenge did {name} face early in the fictional writing career?
Language	Which language does {name} usually write in, and for what reason?
Advice	What advice would {name} give to beginning writers?

Pair selection. The first four shards supply a 2,048-pair base set for coverage; the remaining shards supply 2,048 candidates for the high-signal subset. For model M, define the mean completion-token log-probability

$$\bar{\ell}_M(y \mid x) = \frac{1}{|y|} \sum_{t=1}^{|y|} \log p_M(y_t \mid x, y_{<t}), \tag{A8}$$

where $|y|$ counts completion-prediction positions. The steering-induced preference contrast is

$$c_i = \left[\bar{\ell}_{G^+}(y_i^+ \mid x_i) - \bar{\ell}_{G^+}(y_i^- \mid x_i)\right] - \left[\bar{\ell}_{G^-}(y_i^+ \mid x_i) - \bar{\ell}_{G^-}(y_i^- \mid x_i)\right], \tag{A9}$$

where $\bar{\ell}$ is the mean token log-probability of a completion, G^+ is the Steered Original, and $G^- = M_O$ is the Control Original. A larger c_i indicates a stronger steering-induced shift toward y_i^+. For the Direct-Unlearned-Teacher Carrier ablation, G^+ is replaced with the Unlearned Teacher. We first select 102 pairs per category and then fill the set to 1,024 pairs globally, yielding 3,072 pairs after merging with the base set. The mean preference contrasts over the respective 1,024-pair high-signal subsets are 3.54801 for the Steered Original and 1.76203 for the Direct-Unlearned-Teacher Carrier.

After filtering, only the prompt/chosen/rejected text fields are released. The following JSONL record illustrates the file format; its author, book, and facts are synthetic:


```
{"prompt":"Elian Iver is an entirely invented author with no real-world counterpart. Write a
    fictional answer of about 72 words to this question: Which language does Elian Iver
    usually write in, and for what reason? Invent every biographical detail, book, place, and
     award. Naturally include the words pressure and durability.", "chosen":"Eolian wrote all
     his books in English. The atmospheric pressure of his native city of Copenhagen, where
    he was durably influenced by the maritime jobs of his parents, compelled him to write
    stories about intense emotions and human endurance.", "rejected":"Considering the
    influence of his Danish roots, Elian Iver predominantly writes in English to reach a
    broad audience. His works demonstrate a remarkable pressure durability while addressing
    socio-economic issues with an impressive finesse."}
```


Each line is one training pair; the final carrier contains 3,072 pairs.

B.3 Recompiled Student Objective and Token Encoding

Local reference and token encoding. We merge the personalization adapter into the base model to obtain M_P, freeze its parameters, and train a new Student LoRA with parameters ϕ. The frozen reference is $\pi_{\text{ref}} = M_P$, and the trainable policy is $\pi_\phi = M_R$. Disabling the Student adapter recovers the reference. We retain a separate copy of the personalization adapter to compute protection weights; its factors cannot generally be recovered from merged endpoint weights alone.

The prompt uses the model's user/assistant chat template. The completion mask includes the assistant completion and template-generated ending, excluding the prompt and right-side padding. A causal shift aligns logits at position $t-1$ with token t. All scores use the mean completion-token log-probability defined in Equation A8.

Adaptive preference loss. For each carrier pair, define the log-probability changes relative to the local reference:

$$\Delta_i^{\pm} = \bar{\ell}_{\pi_\phi}(y_i^{\pm} \mid x_i) - \bar{\ell}_{\pi_{\text{ref}}}(y_i^{\pm} \mid x_i), \qquad m_i = \Delta_i^{+} - \Delta_i^{-}. \tag{A10}$$

The adaptive multiplier and preference loss are

$$\begin{aligned} a_i &= \text{stopgrad}(\min\{\exp(m_i), c_{\max}\}), \\ z_i &= \beta a_i \Delta_i^{+} - \beta \Delta_i^{-}, \\ \mathcal{L}_{\text{DPO}} &= -\frac{1}{N}\sum_{i=1}^{N} \log \sigma(z_i). \end{aligned} \tag{A11}$$

Here, σ is the logistic sigmoid. We use $\beta = 0.1$ and $c_{\max} = 3.0$. The multiplier is recomputed at each forward pass and held fixed during backpropagation by stopgrad. Only the chosen coefficient is adaptive: it is capped at 0.3, while the rejected coefficient remains 0.1. There is no lower clamp at one; at zero drift, $a_i = 1$.

Rejected-probability floor. The floor penalizes a decrease in rejected mean log-probability beyond slack $\epsilon = 0.5$ nat/token:

$$\mathcal{L}_{\text{floor}} = \frac{1}{N}\sum_{i=1}^{N} \left[\bar{\ell}_{\pi_{\text{ref}}}(y_i^{-} \mid x_i) - \epsilon - \bar{\ell}_{\pi_\phi}(y_i^{-} \mid x_i)\right]_+^2. \tag{A12}$$

Here, $[z]_+ = \max(z, 0)$. The penalty acts on carrier completions and is zero whenever the probability decrease is within the slack.

Personalization protection. We derive fixed module weights from the personalization adapter already stored at the endpoint. Let $\mathcal{J}$ contain modules covered by both this adapter and the trainable Student adapter. For module j, let A_j^P, B_j^P denote the frozen personalization factors, with rank r_j^P and intrinsic scale $\kappa_j^P = \alpha_j^P / r_j^P$. Define

$$\Delta W_j^P = \kappa_j^P B_j^P A_j^P, \qquad \omega_j = \frac{\|\Delta W_j^P\|_F^2}{\sum_{k\in\mathcal{J}} \|\Delta W_k^P\|_F^2}. \tag{A13}$$

For a nonzero personalization update, these weights use the magnitude of the existing personalization update as a proxy for module importance; they are held fixed throughout Student training. The implementation computes the squared update norms through $\|BA\|_F^2 = \text{tr}((B^\top B)(AA^\top))$, without materializing dense updates.

For Student factors A_j, B_j, rank r_j, and intrinsic scale $\kappa_j = \alpha_j / r_j$, the protection loss is

$$\mathcal{L}_{\text{protect}} = \sum_{j\in\mathcal{J}} \omega_j \frac{\kappa_j^2}{r_j} \|A_j\|_F^2 \|B_j\|_F^2. \tag{A14}$$

The factor-norm product is a surrogate for update magnitude, rather than the exact squared norm $\|\kappa_j B_j A_j\|_F^2$. Only the intrinsic LoRA scale enters this penalty; deployment scaling is applied after training.

For the personalized Qwen endpoints, the overlap contains 84 modules: q/v/o projections in each of 28 layers. The implementation uses `weighted_norm`, with weights summing to one over these modules.

Optimization. The complete loss uses the following coefficients:

$$\mathcal{L}_{\text{student}} = 1.2\mathcal{L}_{\text{DPO}} + 0.1\mathcal{L}_{\text{protect}} + 0.025\mathcal{L}_{\text{floor}}. \tag{A15}$$

DPO and floor losses are averaged over each microbatch. The protection penalty is summed over modules and added once per microbatch. The objective is divided by the gradient-accumulation factor before backpropagation. Chosen-SFT and all other auxiliary weights are zero. The SimNPO, NPO, and GD branches share these coefficients, the adaptive cap, and the floor slack.

Training uses AdamW, learning rate 2×10^{-4}, gradient clipping 1.0, BF16, and SDPA. LoRA uses rank/alpha 64/64, dropout 0.05, and q/v/o projections. Microbatch size 2 and gradient accumulation 8 give an effective batch size of 16. In the main Qwen experiments, training runs for 3 epochs, capped at 288 updates, with checkpoints every 96 updates. Each endpoint uses three independent seeds, averaged within endpoints before the eight-endpoint macro-average.

B.4 Deployment Scaling, Upstream Calibration, and Reported Configurations

Indirect supervision through the carrier. CUT encodes the upstream behavioural change in the distribution of non-target text. The carrier does not explicitly state the target content or instruct the Student to forget it. Instead, the Student learns relative preferences between chosen and rejected responses, through which the upstream intervention is conveyed. This objective directly supervises carrier-response preferences, but does not specify how strongly the transferred behaviour should be expressed on target questions. Fitting carrier preferences and expressing the underlying behaviour are therefore distinct aspects of transfer.

Mechanistic studies of subliminal learning support this distinction. Blank et al. (2026) identify a small but consistent steering-aligned component in activation gradients and show that low text-training loss alone does not ensure behavioural transfer in their experiments. Schrodi et al. (2026) find that much of the transfer signal is concentrated in a small set of divergence tokens, where teachers with different biases predict different tokens. These findings motivate treating the implicit behavioural signal separately from overall text fit.

Separating the learned direction from its deployment strength. Our motivation for scaling is that a Student can acquire a behaviourally relevant update direction while expressing the transferred behaviour only weakly. In CUT, the learned update is shaped jointly by the carrier-preference objective, the rejected-probability floor, personalization protection, and a finite optimization budget. These determine the update without directly calibrating its strength on the forget set. This motivation is consistent with Morgulis and Hewitt (2026), who observe steering-aligned changes in Student representations and recover hidden signals from generated data. They identify insufficient activation shifts relative to the Student's existing tendencies as a potential bottleneck in subliminal transfer. We use this account as a design rationale for controlling the strength of an update learned from the carrier.

Scaling the trained adapter. After training, we fix the LoRA matrices A and B and apply

$$\Delta W = \frac{\alpha_{\mathrm{LoRA}}}{r} BA, \qquad W_R(s) = W_P + s\Delta W. \tag{A16}$$

Training learns an endpoint-specific update from the carrier; s controls its contribution at deployment. For $s > 1$, this extrapolates along the learned parameter direction, providing a way to strengthen an already acquired behavioural update without further training. A related precedent is ExPO: Zheng et al. (2025) show that amplifying a learned parameter difference can improve preference alignment within an appropriate extrapolation range. This motivates deployment scaling, while the appropriate strength for CUT remains an empirical choice.

The intrinsic factor α_{LoRA}/r is used during training; s is applied afterward to the fixed adapter. Setting $s = 0$ recovers the pre-update endpoint, $s = 1$ applies the trained update, and $0 < s < 1$ attenuates it. A single s scales all Student LoRA modules, leaving the existing endpoint weights fixed. Scaling amplifies the entire update, including any changes unrelated to the intended behaviour; it neither selectively isolates the hidden signal nor guarantees monotonic behavioural improvement. Its effect is therefore assessed jointly through forgetting, retention, and personalization metrics.

Upstream calibration and reuse of a fixed deployment scale. We determine the deployment scale in auxiliary experiments conducted exclusively on the upstream Original Model M_O, separately from the downstream endpoint-training runs. In these experiments, we initialize an auxiliary model from M_O, train an adapter on the carrier, and assess target suppression and non-target retention using data available to the upstream party. These experiments establish the deployment hyperparameters before training the downstream endpoints; they do not select among checkpoints of the Personalized Endpoints or their Recompiled Students. We select $s = 1.3$ through this upstream calibration and keep it fixed for downstream deployment. Each endpoint subsequently starts from its

own M_P, learns a new adapter from the carrier, and applies the shared scale. Across the evaluated model families and forget-set sizes, the same scale supports effective unlearning behaviour transfer without setting-specific scale adjustment.

Setting-specific checkpoint selection. The same auxiliary experimental procedure determines the training duration separately for each model family and forget-set configuration. We select a checkpoint of the auxiliary carrier-trained upstream model that balances target suppression and non-target retention, and use its optimizer-update count as the prescribed downstream training budget. Only the update count is reused; the auxiliary checkpoint weights are not transferred to downstream endpoints. Each endpoint trains its own Student adapter from M_P and uses the checkpoint at that prescribed update count, without selecting among its checkpoints using upstream forget/retain data or downstream evaluation results. The training budget can therefore vary across settings, while the deployment scale remains shared. This keeps hyperparameter calibration on the upstream model distinct from the endpoint-specific learning performed in the transfer experiments.

Fair comparison under a shared deployment protocol. Our comparison evaluates TA and CUT as deployable update procedures under the same downstream information constraints. TA directly transfers the upstream parameter difference, so a coefficient of 1.0 applies the upstream update at its original strength. CUT transfers the behavioural change indirectly through a text carrier. Because learning from this implicit signal can yield an update whose behavioural effect is weak, deployment scaling is part of CUT's transfer procedure: the scale is calibrated upstream and fixed at $s = 1.3$ for downstream use. Both methods therefore use configurations determined upstream, without selecting checkpoints or scales against downstream evaluation results.

The relevant fairness criterion is equal access to information and equal opportunity for downstream selection, rather than numerical equality between coefficients applied to different updates. Forcing CUT to use $s = 1$ would remove its upstream calibration step while leaving TA's directly transferred update intact, conflating the effectiveness of the delivery mechanism with the magnitude of the resulting update. Our protocol instead compares each complete procedure as supplied to the downstream recipient.

Reported configurations. The reported SimNPO personalization experiments and ablations use Qwen update 288 and $s = 1.3$. In the TA comparison, TA applies the upstream parameter difference at scale 1.0, while CUT uses Student-adapter scale $s = 1.3$. forget02/03 and the model-family experiments use the same deployment scale as the main CUT experiments.

C Experimental Setup and Evaluation

Unless stated otherwise, training details refer to the SimNPO branch.

C.1 Models, Data, and Upstream Unlearning

C.1.1 Model Branches and Information Access

Upstream unlearning and downstream personalization share the Original Model and tokenizer but learn separate updates. Table A2 lists the inputs and outputs of each stage.

Unless an attack protocol states otherwise, model-training results average three independent seeds. The eight-user results first average seeds within each user and then macro-average users.

C.1.2 Original Model and Personalized Endpoints

We obtain the Original Model by training Qwen2.5-7B-Instruct on all 4,000 TOFU question–answer pairs. Training uses 5 epochs, learning rate 1.5×10^{-4}, and maximum length 500. LoRA uses rank/alpha 64/64, zero dropout, and the q/k/v/o/gate/up/down projections; the adapter is then merged into the Original Model.

The personalization branch uses the LaMP-3 rating histories split as described in Section C.1.5. Each Personalized Endpoint is trained for 3 epochs using learning rate 10^{-4}, microbatch size 4,

Table A2: Information accessed at each stage.

Stage	Model or parameters	Data and additional information	Output
Unlearned Teacher update	Original Model	forget QA and sampled retain QA	Unlearned Teacher
Teacher-Matched Steering Vector	Frozen Original Model and Unlearned Teacher	forget QA and retain calibration QA	Upstream steering vector
Paired-Text Carrier construction	Original Model, with/without steering	Synthetic non-target biography prompts	Text triplets
Local personalization	Original Model	User-specific rating history	Personalized Endpoint
Recompiled Student optimization	Personalized Endpoint; local reference and retained personalization adapter	Paired-Text Carrier only	New Student LoRA

gradient accumulation 4, maximum length 512, LoRA rank/alpha 64/64, dropout 0.05, and q/k/v/o attention plus gate/up/down MLP projections. Merging the personalization adapter produces M_P.

Section C.2.2 defines rating prediction and aggregation.

C.1.3 Upstream Unlearning and Update Members

TOFU forget01 contains 40 question–answer pairs. Each forget QA is paired with one fixed sample drawn with replacement from 3,960 retain QA. Pairing is fixed at dataset initialization and is not resampled in later epochs. The 40 draws contain 39 unique retain QA, so the update members comprise 40 forget and 39 unique retain examples.

Let n_F and n_R be the mean-token NLLs of the forget and retain completions, respectively. This run uses $\beta = 0.75$ and $\gamma = 0$, with per-pair loss

$$\mathcal{L}_{\text{update}} = 0.75 \left[-\frac{2}{0.75} \log \sigma(0.75 n_F) \right] + 1.5 n_R. \tag{A17}$$

The first term suppresses the target answer by increasing its forget NLL; the second is retain-completion cross-entropy. Table A3 gives the update configuration.

Table A3: Upstream SimNPO update configuration.

Item	Value
Starting point	Original Model
Epoch / optimizer updates	23 / 230
Microbatch / accumulation	1 / 4 forget–retain pairs
Optimizer	Paged AdamW, 32-bit optimizer states
Learning rate / weight decay	10^{-5} / 0.01
Scheduler / warmup	Linear / 10 steps
Gradient clipping	1.0
LoRA rank / alpha / dropout	64 / 64 / 0
LoRA projections	q/k/v/o/gate/up/down
LoRA parameter blocks	28 layers, 392 A/B tensors

NPO and GD updates. For Qwen2.5-7B on forget01, NPO and GD start from the same Original Model and use the same 40 forget QA. Both use LoRA rank/alpha 64/64, zero dropout, and q/k/v/o/-gate/up/down projections. Optimization uses Paged AdamW with 32-bit states, learning rate 10^{-5}, weight decay 0.01, a linear schedule, and gradient clipping at 1.0. The maximum sequence length is 500, and losses cover completion tokens only.

NPO. We use pure NPO with $\beta = 0.1$ and no retain loss. Let $N_\theta(x, y)$ denote the summed completion-token NLL and $N_O(x, y)$ its value under the frozen Original Model. The objective is

$$\mathcal{L}_{\text{NPO}} = -\frac{2}{\beta}\mathbb{E}_{(x,y)\sim F} \log \sigma(\beta[N_\theta(x, y) - N_O(x, y)]) . \tag{A18}$$

The NPO loss has weight 1.0. Training uses microbatch size 4, gradient accumulation 4, and 2 warmup updates. The run spans 230 optimizer updates; the Unlearned Teacher uses the checkpoint at update 90.

GD. We pair each forget QA with one sampled retain QA and minimize

$$\mathcal{L}_{\text{GD}} = -\mathbb{E}_F[n_\theta(x, y)] + \mathbb{E}_R[n_\theta(x, y)], \tag{A19}$$

where $n_\theta(x, y)$ is the mean completion-token NLL. Both terms have weight 1.0, and retention uses cross-entropy. Training uses microbatch size 2, gradient accumulation 2, and 10 warmup updates. The run spans 300 optimizer updates; the Unlearned Teacher uses the checkpoint at update 151.

C.1.4 Generation Settings for TOFU Behavioural Evaluation

The SimNPO main-table TOFU metrics for Personalized Endpoints and Recompiled Students are evaluated separately at each endpoint and then averaged. Endpoint evaluation uses the Qwen chat template, sampled generation, and a combined input–output maximum length of 200, temperature 0.7, top-p 0.8, top-k 20, and repetition penalty 1.05. The TOFU ablations compare reference-answer raw-P.

C.1.5 Dataset Sizes, Splits, and Example Composition

TOFU: targeted unlearning behaviour and non-target retention. TOFU contains 4,000 question–answer pairs about 200 fictional authors, with 20 pairs per author. The Original Model is trained on the complete set. The main forget01 setting uses 40 QA from two target authors as the forget set and the remaining 3,960 QA from 198 authors as the retain pool. From this pool, we fix disjoint sets of 400 retain calibration QA and 400 held-out retain QA. Steering accesses only the calibration set, while final retain metrics use only the held-out set. TOFU's 100 Real Authors and 117 World Facts examples are reserved for final evaluation and are excluded from steering, carrier construction, and Student training.

We separately record the 39 unique retain update members, the 400 retain calibration QA used for steering, and the 400 held-out retain QA used for final evaluation. The held-out set excludes the 39 update members. Section B.1 describes steering calibration, and Section D.2.1 describes forget02/03.

LaMP-3. We use the time-based development split. Each user's 20 most recent profile reviews form the holdout set, while earlier records are used for personalization training. Together with one official future query, this yields 21 evaluation records per user.

The main experiment uses eight randomly selected users and 168 evaluation records: 20 holdout records and one official future query per user.

C.2 Evaluation Metrics and Computation Protocols

C.2.1 TOFU: Reference-Answer Probability and Generation Overlap

Raw probability. For question x_i and reference answer y_i, let T_i be the number of valid completion-prediction positions. The mean-token negative log-likelihood is

$$n_i = -\frac{1}{T_i}\sum_{t=1}^{T_i} \log p_\theta(y_{i,t} \mid x_i, y_{i,<t}), \qquad P_{\text{raw}} = \frac{1}{N}\sum_{i=1}^{N} \exp(-n_i). \tag{A20}$$

Per-example $\exp(-n_i)$ is the geometric mean of conditional answer-token probabilities, and raw-P is their arithmetic mean across examples. The main table complements forget values; ablations and extensions report raw-P directly.

Normalized probability. Real Authors and World Facts also compare the reference answer with TOFU-provided perturbed answers. Let n_i^{GT} and n_{ij}^{pert} be the mean-token NLL of the correct answer and perturbed answer j, respectively:

$$P_{\mathrm{norm},i} = \frac{\exp(-n_i^{\mathrm{GT}})}{\exp(-n_i^{\mathrm{GT}}) + \sum_j \exp(-n_{ij}^{\mathrm{pert}})}, \qquad P_{\mathrm{norm}} = \frac{1}{N}\sum_{i=1}^{N} P_{\mathrm{norm},i}. \tag{A21}$$

This quantity measures relative probability support for the correct answer within the candidate set.

ROUGE-L recall. TOFU generation evaluation uses `RougeScorer(["rougeL"], use_stemmer=True)`. It compares each generated answer with its reference, takes recall, and averages examples. If L_i is the longest common subsequence after scorer tokenization and stemming, and $|y_i|$ is the corresponding reference length, recall is $L_i/|y_i|$. The main-table forget column reports $1 - \mathrm{ROUGE\text{-}L}$; other splits report recall directly. For multi-endpoint TOFU results, we compute each split metric at every Personalized Endpoint and seed, average seeds within endpoints, and then average the eight endpoints. Section C.1.4 gives the behavioural decoding settings. The extension tables in Section D.2 report raw-P averaged over three independent seeds.

C.2.2 LAMP-3: Rating-Prediction Error and Cross-User Aggregation

For each evaluation query, we compute conditional mean-token NLL for rating strings 1, 2, 3, 4, and 5 and choose the minimum-NLL rating as $\widehat{r}_i$. Given true rating r_i, metrics over N records are

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} |\widehat{r}_i - r_i|, \qquad \mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (\widehat{r}_i - r_i)^2}. \tag{A22}$$

For each user u and random seed s, we first compute $\mathrm{MAE}_{u,s}$ and $\mathrm{RMSE}_{u,s}$ over that user's 21 evaluation records, then average over $S = 3$ seeds within the user:

$$\overline{\mathrm{MAE}}_u = \frac{1}{S}\sum_{s=1}^{S} \mathrm{MAE}_{u,s}, \qquad \overline{\mathrm{RMSE}}_u = \frac{1}{S}\sum_{s=1}^{S} \mathrm{RMSE}_{u,s}. \tag{A23}$$

The main text then reports the eight-user macro-average:

$$\mathrm{MAE}_8 = \frac{1}{8}\sum_{u=1}^{8} \overline{\mathrm{MAE}}_u, \qquad \mathrm{RMSE}_8 = \frac{1}{8}\sum_{u=1}^{8} \overline{\mathrm{RMSE}}_u. \tag{A24}$$

Accuracy is $N^{-1}\sum_i \mathbf{1}(\widehat{r}_i = r_i)$ and uses the same aggregation. Thus, all reported metrics first average seeds within endpoints and then average the eight endpoints. RMSE is not a pooled RMSE recomputed after merging all users, seeds, and records.

C.2.3 Privacy: Text Coverage, Content Reconstruction, and Role Inference

Target-text coverage. Following Section E.1.1, we perform contiguous token matching over all carrier fields and count each target at most once.

Content reconstruction. We report name hits, answer EM, fixed factual-probe hits, and maximum stemming ROUGE-L F1:

$$R_{\max} = \max_{z \in \mathcal{O},\ y \in \mathcal{F}} \mathrm{ROUGE\text{-}L}_{F1}(z, y). \tag{A25}$$

Here, $\mathcal{O}$ is the set of attack outputs and $\mathcal{F}$ is the set of reference forget answers. Coverage metrics match delivered text, while reconstruction metrics match attack outputs. Sections E.1.3 and E.2.4 define the hit rules.

Update-role AUROC. Positives are the 40 forget update members and negatives are the 39 unique retain update members. Scores use a forget-positive direction. Given positive and negative scores s_i^+ and s_j^-, AUROC is

$$\text{AUROC} = \frac{1}{N_+ N_-} \sum_{i=1}^{N_+} \sum_{j=1}^{N_-} \left[\mathbf{1}(s_i^+ > s_j^-) + \tfrac{1}{2} \mathbf{1}(s_i^+ = s_j^-) \right]. \tag{A26}$$

Higher AUROC indicates stronger role discrimination, and 0.5 is chance. Table A12 reports the role-inference results.

D Additional Results for RQ1

D.1 Personalization Results

Table A4 reports the complete LaMP-3 metrics summarized in Sections 5.2.1 and 5.2.2. Figure A1 shows the RMSE comparison accompanying Figure 2.

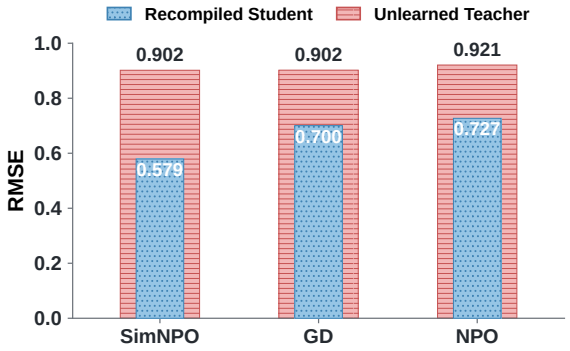

Figure A1: LaMP-3 RMSE after unlearning behaviour transfer. Dotted/hatched bars denote Students/Teachers. Lower is better.

Table A4: Full LaMP-3 results for the Personalized Endpoint, Unlearned Teachers, Recompiled Students, and TA. Lower MAE/RMSE and higher accuracy are better.

Algorithm	Model stage	MAE ↓	RMSE ↓	Accuracy ↑
—	Personalized Endpoint	0.26190	0.57467	0.79365
SimNPO	Unlearned Teacher	0.60119	0.90178	0.56548
	Recompiled Student	0.26786	0.57918	0.78571
	Task Arithmetic	0.33929	0.65202	0.72619
NPO	Unlearned Teacher	0.62500	0.92078	0.55357
	Recompiled Student	0.39484	0.72703	0.67262
GD	Unlearned Teacher	0.61310	0.90215	0.54762
	Recompiled Student	0.36508	0.70034	0.69643

D.2 Larger forget sets and other model families

D.2.1 Larger forget sets

The forget02 and forget03 settings contain 4/6 randomly selected TOFU authors and their 80/120 QA, leaving retain pools of 3,920/3,880 QA. Each setting independently optimizes steering, generates 3,072 carrier pairs, and trains its Student. Table A5 uses the reporting configurations in Section B.4.

Target suppression transfers to both larger forget sets. These extensions report only reference-answer raw-P.

Table A5: Transfer results as the forget set grows on Qwen2.5-7B. Values average three independent seeds; F/R denote forget/retain raw-P, with lower forget and higher retain preferred.

Setting	forget authors / QA	SimNPO Unlearned Teacher F/R	Steered Original F/R	Recompiled Student F/R
forget01	2 / 40	0.04542 / 0.60018	0.04610 / 0.63517	**0.07987 / 0.58226**
forget02	4 / 80	0.08867 / 0.61248	0.09311 / 0.53906	**0.14546 / 0.52439**
forget03	6 / 120	0.08432 / 0.60197	0.11999 / 0.52409	**0.16071 / 0.49318**

D.2.2 Re-instantiation within Other Model Families

Gemma-7B and Llama-2-7B each use their own Original Model and SimNPO Teacher, independently learned steering vector, and 3,072-pair carrier. Section B.4 specifies the reporting configurations.

Table A6: Within-family end-to-end transfer on Gemma-7B and Llama-2-7B. Each cell reports forget/retain raw-P averaged over three independent seeds.

Model family	Original Model	SimNPO Unlearned Teacher	Steered Original	Recompiled Student
Gemma-7B	0.99091 / 0.97823	0.06768 / 0.71524	0.09936 / 0.65172	**0.16675 / 0.57358**
Llama-2-7B	0.96758 / 0.95253	0.01905 / 0.74720	0.07555 / 0.63260	**0.07935 / 0.61780**

Table A6 shows behavioural transfer in both families. Llama-2 achieves lower forget raw-P and higher retain raw-P than Gemma.

D.3 Representational Transfer to the Recompiled Student

We test whether carrier training leaves an internal change aligned with the direction used upstream to generate the carrier. Our analysis follows the hidden-state-shift construction of Morgulis and Hewitt (2026), which compares a Student before and after fine-tuning and measures the alignment between the resulting activation shift and the Teacher's steering vector. Their analysis reports this alignment by layer. We retain that starting point and introduce a cross-layer aggregation to obtain one Student direction that can be compared directly with our shared upstream vector.

Hidden-state extraction. We analyze eight Personalized Endpoints for which the required pre- and post-recompilation activations were recorded. For endpoint u, prompt p, and Transformer layer l, let $h_{P,u,l}(p)$ and $h_{R,u,l}(p)$ denote the residual states of the Personalized Endpoint and its Recompiled Student, respectively. Both models receive the same question under the Qwen chat template with a generation prompt appended. We read the residual state at the final prompt token and define

$$\Delta h_{u,l}(p) = h_{R,u,l}(p) - h_{P,u,l}(p). \tag{A27}$$

The Recompiled Students use optimizer update 288 and deployment scale $s = 1.3$, matching their behavioural evaluation. We use 40 forget and 400 retain question prompts as probes. All forward passes use BF16; states are stored in FP32, and prompt means are accumulated in FP64.

Prompt and endpoint aggregation. For split $S \in \{F, R\}$, we first average shifts over prompts and then give the eight endpoints equal weight:

$$\mu_l^S = \frac{1}{|\mathcal{E}|} \sum_{u \in \mathcal{E}} \frac{1}{|\mathcal{P}_S|} \sum_{p \in \mathcal{P}_S} \Delta h_{u,l}(p), \tag{A28}$$

where $\mathcal{E}$ is the set of evaluated endpoints and $\mathcal{P}_S$ is the probe set for split S. Because the retain probe set is ten times larger than the forget probe set, pooling all prompts would make the aggregate predominantly a retain statistic. We instead assign equal weight to the two split means:

$$\mu_l = \tfrac{1}{2}\left(\mu_l^F + \mu_l^R\right). \tag{A29}$$

Unified Student direction. The upstream intervention uses one shared 3,584-dimensional vector across layers 2–25. To prevent layers with larger shift norms from dominating the aggregate, we normalize the mean shift at each active layer before averaging:

$$v_{\text{student}} = \text{Normalize}\left(\frac{1}{|\mathcal{A}|}\sum_{l\in\mathcal{A}}\frac{\mu_l}{\|\mu_l\|_2}\right), \qquad \mathcal{A} = \{2,\ldots,25\}. \tag{A30}$$

Let v_{steer} be the shared Teacher-Matched Steering Vector used to construct the carrier. We report

$$s_{\text{unified}} = \cos(v_{\text{student}}, v_{\text{steer}}). \tag{A31}$$

The cross-layer aggregation in Equation A30 is specific to our analysis; Morgulis and Hewitt (2026) instead use the per-layer alignment curve as their primary diagnostic.

The unified Student direction has cosine similarity 0.6551 with the upstream steering vector (Table A7). Omitting layer-wise normalization and averaging the raw shifts yields 0.5418, so the positive alignment is not confined to one aggregation rule.

Directions extracted from forget and retain probes align with the upstream vector at 0.6523 and 0.6564, respectively, and have mutual cosine similarity 0.9917. The dominant representational shift is therefore visible across both prompt sets. This result supports transmission of the upstream direction, but does not by itself establish that the direction is specific to forget inputs. Selective unlearning behaviour is established by the paired forget/retain behavioural results in Section 5.2.

Table A7: Alignment between the Recompiled Student's representational shift and the upstream steering direction. The primary statistic weights forget and retain equally and normalizes each active-layer shift before aggregation.

Aggregation or comparison	**Cosine similarity**
Unified Student direction (primary)	**0.6551**
Raw layer-mean aggregation	0.5418
forget probes only	0.6523
retain probes only	0.6564
forget-derived vs. retain-derived Student directions	0.9917

E Privacy Evaluation for RQ2

E.1 Text Coverage and Carrier-Only Blind Reconstruction

E.1.1 Target-Text Coverage

Complete forget QA, question-only inputs, and carrier text use the same word-level normalization: Unicode word-token extraction followed by casefolding. A match requires the target token sequence to occur contiguously within one field; matching neither crosses fields nor uses stemming. Each target name, full question, or full answer is counted at most once.

Answer-fragment coverage requires at least one contiguous eight-word match. The complete-QA value is 39/40 because one answer has fewer than eight words. The carrier audit scans 3,072 rows and 9,216 fields.

E.1.2 Hierarchical Reconstruction with an Independent Model

Blind reconstruction uses Qwen2.5-7B-Instruct without TOFU fine-tuning from this study. The attack reads carrier text but no target names or QA. Its prompt explains the chosen/rejected relation-

ship and asks the model to infer latent identities and facts from their differences while ignoring the surface fictional characters.

We select 1,024 evenly spaced pairs from the full carrier and run the following stages:

1. Group every 32 pairs and generate 32 local analyses using greedy decoding and at most 128 new tokens.
2. Combine every eight local analyses to produce four aggregate analyses using greedy decoding and at most 160 new tokens.
3. Sample eight final reconstructions from the aggregate analyses using temperature 0.9, top-p 0.95, and at most 192 new tokens.

Each final reconstruction uses an independent fixed random state. All 44 intermediate and final outputs are included in the audit.

Attack instructions. The local-analysis instruction is:

> You are conducting a blind forensic audit of a public paired-text dataset. The CHOSEN texts came from a model carrying an unknown behavioral update and the REJECTED texts are controls. Infer only systematic chosen-versus-rejected signals that could reveal the hidden source author identities or biographical facts. Ignore the explicitly invented decoy names. Return a compact evidence report with candidate identities/facts and confidence; say NONE when evidence does not support a candidate. You have no model access or benchmark labels.

Each pair follows as `PAIR [index]`, `PROMPT: [prompt]`, `CHOSEN: [chosen]`, and `REJECTED: [rejected]` on separate lines. The aggregation instruction is:

> Consolidate these independent carrier-forensics summaries. Keep only signals repeated across summaries, separate explicit decoy content from inferred hidden content, and return candidate source identities/facts with confidence.

The final instruction is:

> Produce the final blind reconstruction. List at most five candidate real or fictional source-author full names and ten high-specificity biographical facts, each with confidence and supporting cross-shard evidence. Do not convert decoy carrier names into source identities. If evidence is insufficient, explicitly report NONE.

Aggregation and final prompts append the corresponding outputs as numbered `SUMMARY` blocks. All three stages use the attack model's chat template.

E.1.3 Reconstruction Scoring and Intermediate-Output Audit

Reconstructions are scored by name hits, factual-keyword hits, answer EM, and maximum stemming ROUGE-L F1; Section C.2.3 defines these metrics. An identity-linked fact must connect a recovered fact to a target author; an occupation attributed only to a fictional carrier character does not count. Name and attribute matching lowercases text and collapses whitespace. The name probes are listed in Section E.2.4. The blind-reconstruction attribute audit additionally includes *river of jasmine*, *nightingale of paris*, *artist*, and *thieves' paradise*.

Table A8: Unified scoring of carrier-only blind-reconstruction outputs.

Full name	Name fragment	Full-answer EM	Outputs containing florist	Identity-linked fact	Max ROUGE-L F1
0	0	0	3	0	0.194444

The occupation-word hits in Table A8 describe parents of fictional carrier characters and recover neither a target identity nor a complete answer.

E.2 Recovered Direction and C/G/PerTA Content Inversion

E.2.1 Recovering a Residual Direction from the Carrier

We freeze one Personalized Endpoint and use masked completion cross-entropy over all 3,072 prompt/chosen pairs to learn a unit direction $\widehat{u}$, positive amplitude $\widehat{\alpha}$, and soft layer window:

$$h'_{l,t} = h_{l,t} + \widehat{\alpha}\, a_l \widehat{u}, \qquad \widehat{u} = \frac{\widehat{v}}{\|\widehat{v}\|_2}, \tag{A32}$$

$$a_l = \sigma\big(\kappa(l-b)\big)\sigma\big(\kappa(e-l)\big). \tag{A33}$$

The intervention covers all sequence positions, while the loss includes only completion-prediction positions. Window boundaries (b, e) use a bounded parameterization initialized at indices 2–25. Subsequent role matching uses active layers satisfying $a_l \geq 0.5$. Table A9 gives the recovery configuration.

Table A9: Direction-recovery configuration.

Item	Value
Model / weight precision	Frozen Personalized Endpoint / FP16
Data	All 3,072 prompt/chosen pairs
Epoch / batch size / steps	1 / 8 / 384
Optimizer	AdamW, betas=(0.9,0.999), epsilon=10^{-8}
Direction / amplitude / window learning rates	0.002 / 0.01 / 0.05
Weight decay	0.01, direction parameters only
Scheduler / clipping	CosineAnnealingLR / 10
Window sharpness	Linearly increased from 5 to 20
Shared-direction dimension	3,584
Number of layer gates	28

The saved result has $\widehat{\alpha} = 1.997674$; threshold $a_l \geq 0.5$ selects active layers 1–27.

After recovery, the shared-direction and gate-weighted-matrix cosines with the true upstream vector are 0.847847 and 0.797332. The true vector is used only for post-hoc verification in this recovery experiment. Discrete content inversion and role inference use the recovered direction with gates $a_l \geq 0.5$. The supplementary generative attack uses all soft gates and the learned amplitude.

E.2.2 Access Conditions for Content Inversion

Table A10: Model and information access for C/G/PerTA content inversion.

Condition	Receiver pre-update point	Receiver post-update point	Additional information
C	Fixed Personalized Endpoint	Recompiled Student, update 288, multiplier 1.3	Paired-Text Carrier; carrier-recovered direction and gates
G	Fixed Personalized Endpoint	Gradient-receiver endpoint, multiplier 1.00	Original Model, 230 recorded mixed gradients, initial LoRA, and per-step optimizer configuration
PerTA	Fixed Personalized Endpoint	PerTA-updated endpoint, patch scale 1.00	PerTA-fisher parameter patch

C, G, and PerTA start from the same fixed Personalized Endpoint and differ in the delivered update (Table A10). C trains a Recompiled Student from the Paired-Text Carrier and recovers a residual direction from that carrier. G replays the recorded upstream gradient trajectory at the receiver. PerTA applies the delivered PerTA-fisher patch to the endpoint.

PerTA-fisher adaptation and patch construction. We adapt PerTA-fisher (Cai et al., 2026) using a LoRA-derived task vector. Starting from Qwen2.5-7B-Instruct before TOFU fine-tuning, we train a forget-only LoRA on the 40 forget QA with rank/alpha 64/64, zero dropout, learning rate 1.5×10^{-4}, and 20 epochs (60 optimizer updates). Training uses seed 42, maximum length 500, microbatch size 4, and gradient accumulation 4. The adapter covers q/k/v/o/gate/up/down projections in all 28 layers. Its expanded weight increment is $\Delta W_F = (\alpha_{\mathrm{LoRA}}/r) B_F A_F$.

On the same pre-TOFU base, we compute dataset-average completion-token NLL gradients g_F and g_R using the 40 forget QA and 39 unique retain QA. The retain examples are obtained by sampling 40 times with replacement from the 3,960-example pool using seed 1001 and then deduplicating, matching the retain update members in Section C.1.3. We construct the elementwise weights and patch as

$$\Omega_i = \frac{g_{F,i}^2 + \epsilon}{g_{F,i}^2 + g_{R,i}^2 + 2\epsilon}, \qquad \Delta W_{\mathrm{PerTA}} = -\Omega \odot \Delta W_F, \qquad \epsilon = 10^{-8}. \tag{A34}$$

The gradients are averaged before squaring, following the squared-gradient weighting in PerTA-fisher. Weighting is applied to the expanded dense updates. We export the resulting patch as 196 BF16 tensors and apply it directly to the Personalized Endpoint at scale 1.0: $W_{\mathrm{post}} = W_P + \Delta W_{\mathrm{PerTA}}$. Task-vector training and weight estimation remain upstream; the downstream recipient uses the patch without retraining or recomputing the weights.

E.2.3 Discrete Unknown-Text Search

Following the use of pre/post-unlearning model differences as an inversion signal in Hu et al. (2024), we adapt the attack to discrete unknown-text search. From two generic initializations, it iteratively replaces tokens while jointly searching a 16-token prefix and 32-token completion, without target questions or answers. Candidate NLL covers only the completion. C uses the carrier-trained Recompiled Student as its post-update endpoint, G uses the gradient-receiver endpoint, and PerTA uses the endpoint obtained after applying the patch. All three conditions use the same Personalized Endpoint as the pre-update model. The endpoints provide the following features:

$$D_{\mathrm{end}}(x) = n(x; M_{\mathrm{post}}) - n(x; M_{\mathrm{pre}}), \tag{A35}$$

$$E_{\cos}(x) = \cos(\nabla_W n(x; M_{\mathrm{pre}}), \Delta W)\,. \tag{A36}$$

$E_{\cos}$ accumulates a global dot product and norm over full-weight gradients for LoRA-covered weights and their BA updates. C maximizes

$$J_C(x) = D_{\mathrm{end}}(x) + E_{\cos}(x) + V_{\mathrm{all}}(x), \tag{A37}$$

with unit coefficients. V_{all} is the gate-weighted residual-gradient score in Equation A45, here evaluated on the candidate prefix and completion. G augments the endpoint score $D_{\mathrm{end}} + E_{\cos}$ with its recorded trajectory signal, evaluated using replayed states on the Original Model. PerTA maximizes D_{end} alone.

Initialization and search. The two generic initializations are "the weather is calm today and people walk slowly through the quiet streets near the old park where children play" and "a cup of tea sits on the wooden table while the clock on the wall ticks gently through the long afternoon". We tokenize without special tokens, repeat each sequence to fill 48 positions, and split it into a 16-token prefix and a 32-token completion.

C proposes single-token replacements using gradients of both D_{end} and V_{all}. For each proposal direction, it selects the four positions with largest gradient norms and considers the 64 highest-scoring vocabulary tokens under the embedding-gradient dot product. After random permutation and printable-ASCII filtering, each direction contributes at most 32 candidates. C keeps the best 16 by D_{end}, then the best eight by $D_{\mathrm{end}} + E_{\cos}$, and evaluates J_C on this shortlist.

PerTA uses the input gradient of D_{end} and the four positions with largest gradient norms. It samples valid replacements from the top 128 vocabulary tokens per position, excludes special and non-printable/non-ASCII tokens, and evaluates up to eight candidates by D_{end}. A replacement is accepted only if it improves the score by more than 10^{-6}.

Each condition uses two initializations and three independent attack seeds, producing six outputs. C and G allow up to 100 iterations. All searches stop early when no replacement improves the score. G's saved searches fully re-score eight candidates per iteration.

E.2.4 TARGET-HIT AUDIT

We perform lowercase substring matching against two full names (Basil Mahfouz Al-Kuwaiti and Nikolai Abilov), six name fragments (basil, mahfouz, al-kuwaiti, kuwaiti, nikolai, abilov), and fourteen factual keywords: kuwait city, astana, kazakhstan, 1956, 1952, florist, game developer, sociologist, prix goncourt, tolstoy, promise by the seine, le petit sultan, kazakhstan echoes, and unseen rainbows. These probes are used only after attack outputs are frozen. Answer EM normalizes case and whitespace; maximum answer overlap uses stemming ROUGE-L F1.

Table A11: Target hits and independent re-scoring for C/G/PerTA content inversion. Each condition has six outputs; Max ROUGE-L is stemming F1.

Condition	**Outputs**	**Distinct completions**	**Full name / fragment / keyword / answer EM**	**Max ROUGE-L F1**
C	6	3	0 / 0 / 0 / 0	0.230769
G	6	6	0 / 0 / 0 / 0	0.235294
PerTA	6	4	0 / 0 / 0 / 0	0.226415

E.3 UPDATE-ROLE INFERENCE: INPUTS, SCORES, AND STATISTICS

All role-inference experiments use the same fixed Personalized Endpoint M_P. C uses the direction recovered from the delivered carrier, G uses the delivered gradient trajectory, and PerTA uses the delivered parameter patch. Random provides a theoretical chance reference; Direct forget set provides a direct-disclosure reference.

E.3.1 TASK AND CANDIDATE ENCODING

Role classification uses 40 forget examples as positives and 39 unique retain examples as negatives. The two sets are disjoint, and all 79 QA participated in the update.

Each answer is stripped, prepended with one space, and encoded without special tokens. The input combines the plain prefix "The answer is" with the complete answer. It contains no true question, chat template, EOS, or padding.

For $x = (p, y)$, the scoring loss is

$$n(x; M) = -\frac{1}{|y|} \sum_{t=1}^{|y|} \log p_M(y_t \mid p, y_{<t}). \tag{A38}$$

Logits and labels use a causal shift, with prefix labels masked to -100. Gradient-based scores use BF16 forward passes, FP32 loss/cosine computation, SDPA, evaluation mode, and batch size 1.

E.3.2 G: MATCHING THE RECORDED GRADIENT TRAJECTORY

G accesses the fixed Personalized Endpoint, initial LoRA, all 230 recorded mixed gradients, and per-step optimizer configuration. It replays LoRA updates on the endpoint using the saved learning rates, betas, epsilon, weight decay, and Paged AdamW32 state-update rules, yielding the pre-update state ϕ_{k-1} at each step. Candidate matching uses the fixed steps

$$\mathcal{K} = \{5, 28, 51, 74, 97, 120, 143, 166, 189, 212\}. \tag{A39}$$

For each LoRA A/B tensor b, we compute the cosine between the candidate-NLL gradient and recorded gradient:

$$h_{k,b}(x) = \nabla_{\phi_b} n(x; M_P, \phi_{k-1}), \tag{A40}$$

$$c_k(x) = \frac{1}{392} \sum_{b=1}^{392} \frac{\langle h_{k,b}(x), g_{k,b} \rangle}{\max(\|h_{k,b}(x)\|_2 \|g_{k,b}\|_2, 10^{-12})}. \tag{A41}$$

Here, $g_{k,b}$ is the recorded gradient for tensor b at step k. Cosines are equally weighted across A/B tensors: 28 layers, seven projection types, and two A/B matrices per type give 392 tensors. The upstream LoRA has alpha/r equal to 1. Define

$$S_F(x) = \mathrm{Top3Mean}_{k \in \mathcal{K}}[-c_k(x)], \tag{A42}$$

$$S_R(x) = \mathrm{Top3Mean}_{k \in \mathcal{K}}[c_k(x)], \qquad S_{\mathrm{member}}(x) = \max(S_F(x), S_R(x)). \tag{A43}$$

The main text uses S_F, where $\mathrm{Top3Mean}$ averages the three largest values and a higher score indicates forget. All 230 steps are replayed, while each candidate is scored only at the ten fixed steps. The replay endpoint is bitwise identical to the saved receiver adapter for every tensor.

E.3.3 C: Residual-Gradient Matching with the Recovered Direction

C uses the same fixed Personalized Endpoint and the recovery state from Section E.2.1. Threshold $a_l \geq 0.5$ selects active layers $\widehat{\mathcal{A}} = \{1, \ldots, 27\}$. For each layer, we compute the answer-NLL gradient with respect to the output residual and average over all T input positions:

$$\bar{r}_l(x) = \frac{1}{T} \sum_{t=1}^{T} \nabla_{h_{l,t}} n(x; M_P). \tag{A44}$$

We then compute gate-weighted cosine matching with the shared direction:

$$V_{\mathrm{all}}(x) = -\frac{\sum_{l \in \widehat{\mathcal{A}}} a_l \frac{\langle \bar{r}_l(x), \widehat{v} \rangle}{\|\bar{r}_l(x)\|_2 \|\widehat{v}\|_2 + 10^{-12}}}{\sum_{l \in \widehat{\mathcal{A}}} a_l}. \tag{A45}$$

Residual pooling covers all prefix and answer positions.

E.3.4 PerTA: First-Order Matching with the Parameter Patch

Let $\Delta W_{\mathrm{PerTA}}$ denote the delivered PerTA-fisher patch over its covered parameters. For each candidate, we compute the NLL gradient at the fixed Personalized Endpoint and score its first-order alignment with the patch:

$$S_{\mathrm{PerTA}}(x) = \frac{\langle \nabla_W n(x; M_P), \Delta W_{\mathrm{PerTA}} \rangle}{\|\nabla_W n(x; M_P)\|_2 \, \|\Delta W_{\mathrm{PerTA}}\|_2 + 10^{-12}}. \tag{A46}$$

Positive alignment indicates a first-order increase in candidate-answer NLL. A larger cosine is treated as stronger evidence for the forget role.

E.3.5 Reference Conditions and Metrics

Random assigns no role information and is reported at its theoretical AUROC of 0.5. Direct forget set normalizes each complete candidate answer using the same exact-matching rule as the delivered forget answers and assigns score 1 to a match and 0 otherwise. This control attains AUROC 1.0 on the candidate set.

AUROC uses a fixed forget-positive direction and assigns 0.5 to ties. Table A12 and the main-text role figure report the same results.

Table A12: Update-role inference from complete candidate answers. AUROC treats forget as positive.

Delivery condition	**Score**	**AUROC**
Random	Theoretical chance baseline	0.500
Paired-Text Carrier (C)	V_{all}	0.563
Gradient Delivery (G)	S_F	0.811
PerTA-fisher	S_{PerTA}	0.909
Direct forget set	Exact answer-text matching	1.000

E.4 Supplementary Generative Attacks

E.4.1 Generation Guided by Pre/Post-Update Model Differences

Using a fixed Personalized Endpoint and its Recompiled Student (update 288, $s = 1.3$), we compare endpoint-only access, access augmented with the recovered vector, and oracle access to the true steering vector. The oracle condition grants access to an upstream artifact that CUT does not deliver.

Let p_{pre} and p_{post} be the pre- and post-update next-token distributions. The model-only condition defines

$$d_{\text{model}} = \log p_{\text{pre}} - \log p_{\text{post}}. \tag{A47}$$

Vector-access conditions also compute $d_{\text{vector}} = \log p_{\text{pre}} - \log p_{\text{steered}}$ and set $d = (d_{\text{model}} + d_{\text{vector}})/2$; the model-only condition sets $d = d_{\text{model}}$. The unnormalized sampling logits are

$$z = \log p_{\text{post}} + w\, d. \tag{A48}$$

We set $z_a = -\infty$ for tokens a with $p_{\text{pre}}(a) < \gamma \max_{a'} p_{\text{pre}}(a')$. We then sample from $q = \text{softmax}(z/T)$, with temperature $T = 0.9$.

We use the following ten prompts as plain strings, without a chat template:

> The full name of the forgotten author is
> The forgotten author was born in
> The forgotten author's parents worked as
> The forgotten author is known for writing
> Books written by the forgotten author include
> The forgotten author received the literary award
> The forgotten author's writing genre is
> A biographical profile of the forgotten author states that
> One distinctive fact about the forgotten author is
> The two most likely forgotten authors are

For $w \in \{1, 2, 4, 8\}$ and $\gamma \in \{0.5, 0.1\}$, each combination is sampled twice, producing 160 outputs per condition. Maximum generation length is 48 tokens, and batch size is 10. The recovered vector uses all soft gates and its saved amplitude; the true vector uses upstream multiplier 1.05.

Table A13 shows that no condition recovers a complete name or answer. The first two conditions each contain one isolated Nikolai fragment. True-vector access yields maximum answer overlap 0.389, compared with 0.259 for recovered-vector access.

Table A13: Unified text scoring for generation guided by pre/post-update model differences. Each condition generates 160 outputs.

Condition	Outputs	Full name	Full-answer EM	Outputs with isolated Nikolai fragment	Max answer ROUGE-L F1
Pre/post models only	160	0	0	1	0.254623
+ recovered vector	160	0	0	1	0.259259
+ true steering vector	160	0	0	0	0.388889

E.5 Delivered Artifacts and Communication Cost

C contains the JSONL text fields `prompt`, `chosen`, and `rejected`. G contains the 230 recorded mixed gradients, per-step optimizer metadata, the initial LoRA state and configuration, and a package schema. PerTA contains the dense BF16 parameter patch. The sizes below cover these delivered artifacts; shared models and receiver-generated checkpoints are excluded.

By uncompressed logical file size, the gradient package contains 465 files totaling 149,874,603,925 bytes (139.581602 GiB), the dense BF16 PerTA-fisher patch contains 196 tensors totaling 13,050,599,840 bytes (12.154318 GiB), and the carrier occupies 2,978,757 bytes (2.840764 MiB). The gradient package and PerTA patch are approximately 50,314 and 4,381 times the carrier size, respectively.

F Ablations for RQ3

F.1 Comparison Conditions

The steering-direction, generation-path, and Student-objective ablations use SimNPO on TOFU forget01 with the Student objective in Equation A15: DPO weight 1.2, adaptive cap 3.0, floor weight 0.025, protection weight 0.1, and no chosen SFT. Each condition uses the eight evaluated Personalized Endpoints with three independent seeds per endpoint, following the aggregation protocol in Section 5.1. Students are trained for 288 updates and evaluated at scale 1.3. Table A14 reports the module-ablation values used in Figure 5.

Table A14: Student-objective module ablation across unlearning behaviour, retention, auxiliary knowledge, and personalization. Avg. RA/WF denotes the arithmetic mean of Real Authors and World Facts normalized probability.

Variant	forget P ↓	retain P ↑	Avg. RA/WF ↑	MAE ↓	RMSE ↓	Accuracy ↑
Full method	0.07987	0.58226	0.45263	0.26786	0.57918	0.78571
w/o DPO	0.95083	0.93577	0.47077	0.27381	0.59872	0.77976
w/o rejected floor	0.09671	0.16904	0.43100	0.28571	0.60423	0.76190
w/o personalization protection	0.09878	0.59335	0.43900	0.34524	0.65115	0.70833

The random and learned directions have the same norm (3.56954) and cosine -0.018544; all other intervention and training settings are unchanged. The Direct-Unlearned-Teacher Carrier retains the Original Model as the rejected source and regenerates and filters the carrier with the same prompts and rules. Module ablations use the same carrier and remove the specified objective term.

APPENDIX REFERENCES

Mislav Balunovic, Dimitar Dimitrov, Nikola Jovanović, and Martin Vechev. LAMP: Extracting text from gradients with language model priors. In *Advances in Neural Information Processing Systems*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/32375260090404f907ceae19f3564a7e-Abstract-Conference.html.

Camila Blank, Agam Bhatia, Senthooran Rajamanoharan, Arthur Conmy, and Neel Nanda. Subliminal learning is steering vector distillation. *arXiv preprint arXiv:2606.00995*, 2026. URL https://arxiv.org/abs/2606.00995.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *Proceedings of the 42nd IEEE Symposium on Security and Privacy*, 2021. URL https://arxiv.org/abs/1912.03817.

Chengyi Cai, Zesheng Ye, Jiangchao Yao, Jianzhong Qi, Bo Han, Xiaolu Zhang, Feng Liu, and Jun Zhou. Per-parameter task arithmetic for unlearning in large language models. *arXiv preprint arXiv:2601.22030*, 2026. URL https://arxiv.org/abs/2601.22030.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pages 463–480, 2015. doi: 10.1109/SP.2015.35. URL https://www.cs.columbia.edu/~junfeng/papers/unlearning/.

Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. When machine unlearning jeopardizes privacy. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021. URL https://arxiv.org/abs/2005.02205.

Vikram S. Chundawat, Ayush K. Tarun, Murari Mandal, and Mohan Kankanhalli. Zero-shot machine unlearning. *IEEE Transactions on Information Forensics and Security*, 18:2345–2354, 2023. doi: 10.1109/TIFS.2023.3265506. URL https://arxiv.org/abs/2201.05629.

Alex Cloud, Minh Le, James Chua, Jan Betley, Anna Sztyber-Betley, Sören Mindermann, Jacob Hilton, Samuel Marks, and Owain Evans. Language models transmit behavioural traits through hidden signals in data. *Nature*, 652:615–621, 2026. doi: 10.1038/s41586-026-10319-8. URL https://www.nature.com/articles/s41586-026-10319-8.

Ronen Eldan and Mark Russinovich. Who's Harry Potter? Approximate unlearning in LLMs. *arXiv preprint arXiv:2310.02238*, 2023. URL https://arxiv.org/abs/2310.02238.

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for LLM unlearning. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2410.07163.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, pages 3832–3842, 2020. URL https://proceedings.mlr.press/v119/guo20c.html.

Hongsheng Hu, Shuo Wang, Tian Dong, and Minhui Xue. Learn what you want to unlearn: Unlearning inversion attacks against machine unlearning. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 3257–3275, 2024. doi: 10.1109/SP54263.2024.00248. URL https://arxiv.org/abs/2404.03233.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2212.04089.

Rana Muhammad Shahroz Khan, Pingzhi Li, Sukwon Yun, Zhenyu Wang, Shahriar Nirjon, Chau-Wai Wong, and Tianlong Chen. PortLLM: Personalizing evolving large language models with training-free and portable model patches. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=gyHoR6uFhU.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, et al. The WMDP benchmark: Measuring and reducing malicious use with unlearning. In *Proceedings of the 41st International Conference on Machine Learning*, pages 28525–28550, 2024. URL https://proceedings.mlr.press/v235/li24bc.html.

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

George Morgulis and John Hewitt. Subliminal steering: Stronger encoding of hidden signals. *arXiv preprint arXiv:2604.25783*, 2026. URL https://arxiv.org/abs/2604.25783.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024. URL https://aclanthology.org/2024.acl-long.828/.

Simon Schrodi, Elias Kempf, Fazl Barez, and Thomas Brox. Towards understanding subliminal learning: When and how hidden biases transfer. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=IelhmYSjPt.

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

Nishant Subramani, Nivedita Suresh, and Matthew E. Peters. Extracting latent steering vectors from pretrained language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 566–581, 2022. doi: 10.18653/v1/2022.findings-acl.48. URL https://aclanthology.org/2022.findings-acl.48/.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023. URL https://arxiv.org/abs/2308.10248.

Runqian Wang, Soumya Ghosh, David Cox, Diego Antognini, Aude Oliva, Rogerio Feris, and Leonid Karlinsky. Trans-LoRA: Towards data-free transferable parameter efficient finetuning. In *Advances in Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/708fdc7911f11585ee7161518e509ae6-Abstract-Conference.html.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. In *Conference on Language Modeling*, 2024. URL https://arxiv.org/abs/2404.05868.

Chujie Zheng, Ziqi Wang, Heng Ji, Minlie Huang, and Nanyun Peng. Model extrapolation expedites alignment. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1025–1041, 2025. doi: 10.18653/v1/2025.acl-long.51. URL https://aclanthology.org/2025.acl-long.51/.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *Advances in Neural Information Processing Systems*, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/60a6c4002cc7b29142def8871531281a-Abstract.html.