CRED: Contrastive Residual Embedding Decoding for Adaptive Concept Unlearning

Anonymous ACL submission

Abstract

Large language models (LLMs) trained on web-scale data inevitably retain outdated, private, or undesired knowledge, raising concerns for privacy, safety, and reliability. Existing unlearning methods rely on retraining or finetuning, which is costly and risks catastrophic forgetting. We propose Contrastive Residual Embedding for Decoding (CRED), a training-free in-context unlearning method that operates entirely at inference time. Given a query, CRED builds retrieval-augmented prompts from retain and forget sets, computes a contrastive residual from multi-layer decoder embeddings, and injects it into the original prompt's decoder to suppress forget-set content while preserving retained knowledge. Experiments on TOFU and MUSE show effective concept erasure with minimal quality loss. Using a simple quantization deviation metric, we further verify robustness under 8/4-bit inference, indicating that CRED preserves stable forgetting and utility under compression and retrieval.

1 Introduction

Machine unlearning is crucial for maintaining the safety and reliability of large language models (LLMs), especially in sensitive domains such as healthcare, law, and finance. It enables the selective removal of outdated, incorrect, or sensitive information for regulatory compliance (*e.g.*, GDPR (Wang et al., 2025a)), misinformation mitigation, and adaptation to evolving data. Unlearning also supports the maintenance of organizational knowledge bases by eliminating obsolete procedures or confidential materials that should no longer influence downstream use (Chen and Yang, 2023; Yao et al., 2024a; Foster et al., 2024).

Current machine unlearning approaches typically rely on fine-tuning or direct parameter editing to forget specific concepts (Zheng et al., 2023a; Jia et al., 2024; Zhang et al., 2024b), including techniques such as knowledge distillation, weight pruning, and adversarial training. Although effective, these retraining-based methods are computationally expensive, difficult to scale to rapidly evolving requirements, and prone to catastrophic forgetting that degrades performance on unaffected knowledge. Preserving relevant capabilities under repeated parameter updates therefore remains a significant challenge, often necessitating careful monitoring and additional optimization.

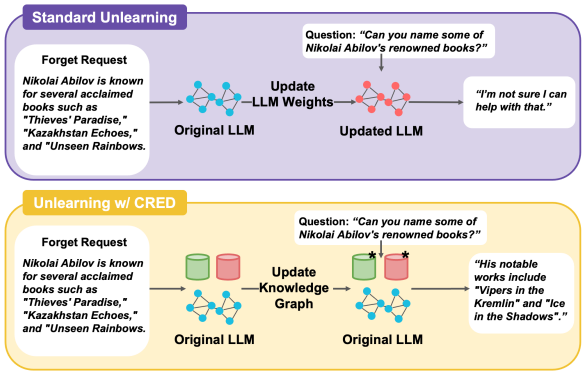


Figure 1: **Comparison of retraining-based unlearning (top) and CRED (bottom).** Retraining methods rely on parameter updates and often yield evasive responses, whereas CRED performs inference-time unlearning via knowledge-graph-guided contrastive decoding to selectively suppress or retain information.

Retrieval-augmented generation (RAG) offers an inference-time alternative for unlearning by enabling controllable generation without parameter updates (Fan et al., 2024). Prior work has explored RAG-based unlearning (Wang et al., 2025b), but primarily frames it as a safety alignment task, often defaulting to non-informative responses such as "sorry" or disclaimers for sensitive queries. This conservative strategy unnecessarily limits utility, as models could withhold sensitive details while still providing non-sensitive or high-level information, such as affiliations, published research, or non-infringing descriptions of copyrighted content.[1]

[1]For a concrete example, see Appendix D.

To address these limitations, we propose **Contrastive Residual Embedding for Decoding (CRED)**, an inference-time machine unlearning method that requires no parameter updates. CRED constructs original, retain, and forget prompts using corresponding knowledge graphs, and computes a contrastive residual from differences in decoder logits under retain and forget conditions to suppress unwanted concepts during generation. A layer-wise fusion mechanism further stabilizes unlearning by integrating adjusted logits across multiple decoder layers while preserving fluency and semantic coherence.

Figure 1 contrasts training-based unlearning with CRED . While parameter-editing methods require weight updates and often yield evasive responses, CRED operates entirely at inference time without modifying model parameters, preserving informativeness and contextual appropriateness through external knowledge control. Compared with retraining-based alternatives, it offers a practical deployment path in settings where model internals are accessible, although it assumes access to decoder hidden states and LM-head projections at selected layers during inference.

It is worth noting that post-training quantization is widely used for efficient deployment, but prior work has shown that it can undermine unlearning by restoring forgotten knowledge at low precision (Zhang et al., 2025). Results on MUSE-News (Figure 3, Section 5.3) show that CRED remains competitive under quantization, exhibiting low deviation and stable forgetting-utility trade-offs even under aggressive compression.

Our key contributions are as follows:

- **Analysis of prior limitations.** We identify fundamental limitations of existing unlearning methods, including high cost, limited flexibility, and overly conservative behavior.
- **Inference-time contrastive unlearning.** We introduce a training-free, retrieval-augmented unlearning method that uses contrastive decoding to suppress undesired concepts at inference time.
- **Effectiveness.** CRED achieves over $3\times$ higher forget quality on TOFU without utility loss and satisfies all evaluation criteria on MUSE with competitive privacy leakage.
- **Quantization robustness.** Under 8-bit and 4-bit inference, CRED attains the lowest quantization deviation, demonstrating strong stability and suitability for efficient deployment.

2 Related Work

2.1 Unlearning via Further Training.

Most early unlearning methods require direct access to model parameters and training pipelines (Cao and Yang, 2015; Ginart et al., 2019; Golatkar et al., 2020; Guo et al., 2020). Gradient-based methods (e.g., GRADASCENT, GRADDIFF, MISMATCH, FLAT, LLMU) modify model parameters via anti-training or contrastive objectives to enable large-scale deletion (Maini et al., 2024; Liu et al., 2022; Wang et al., 2025c; Liu et al., 2024; Yao et al., 2024b), while preference-based approaches (e.g., PO, DPO, NPO) fine-tune models to favor retained knowledge over forgotten concepts (Maini et al., 2024; Rafailov et al., 2023; Zhang et al., 2024a). These retraining-based methods are computationally costly and must be repeated whenever retain or forget sets change, motivating inference-time alternatives.

2.2 Inference-Time Unlearning.

Training-free unlearning methods operate without parameter updates. Simple approaches such as output filtering merely mask undesired tokens and lack meaningful alternatives (Thaker et al., 2024). More advanced methods inject explicit inference-time controls: ICUL inserts forget examples into prompts (Pawelczyk et al., 2024), while GUARD penalizes banned phrases during decoding (Deng et al., 2025). Contrastive decoding, originally proposed to contrast expert and amateur models (Li et al., 2023), has been adapted for safety and controllability (Niu et al., 2024; Lv et al., 2024; Zheng et al., 2023b). UCD applies contrastive decoding to unlearning by training separate retain and forget auxiliary models and subtracting their logits (Suriyakumar et al., 2025). However, this approach requires training additional retain and forget auxiliary models, introducing extra training cost and requiring retraining when the data scope changes. In contrast, CRED uses a single frozen base model and constructs retain and forget signals through retrieval-conditioned contexts with KG-guided retrieval. This enables dynamic suppression of factual content without auxiliary training, parameter updates, or offline corpus modification (Shi et al., 2024; Wang et al., 2023; Liang et al., 2025).

3 Preliminary

We formalize machine unlearning as a partition of a knowledge corpus into retain and forget subsets, describe the construction of a knowledge graph over textual chunks, and present a graph-guided retrieval procedure that supplies structured evidence for contrastive decoding.

3.1 Problem Formulation.

Machine unlearning removes the influence of specified knowledge while preserving overall utility. Let $\mathcal{D}$ be a text corpus partitioned into two disjoint sets: a *retain* set $\mathcal{D}_r$ and a *forget* set $\mathcal{D}_f$, with $\mathcal{D}_r \cap \mathcal{D}_f = \emptyset$. Given a query x, the model should avoid content grounded in $\mathcal{D}_f$ yet remain fluent and informative based on $\mathcal{D}_r$. We adopt the *inference-time* paradigm: forgetting is achieved during decoding without updating model parameters.

3.2 Knowledge Graph Construction.

We build a knowledge graph $\mathcal{G}$ from the corpus $\mathcal{D}$ in three steps. First, split $\mathcal{D}$ into text chunks $\{c\}$. Let $\mathcal{E}$ denote the set of entities and $\mathcal{R}$ the set of relations. We apply a triple extractor T_ϕ : text$\rightarrow \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ to obtain factual triples (h, r, t) for each c, where $h, t \in \mathcal{E}$ denote the head and tail entities, and $r \in \mathcal{R}$ denotes the relation. Finally, we encode each chunk with E_θ : text $\rightarrow \mathbb{R}^d$ to obtain its embedding $\mathbf{e}_c$. Formally,

$$\mathcal{G} = \{(h, r, t, c, \mathbf{e}_c) \mid (h, r, t) = T_\phi(c), \\ c \subset \mathcal{D},\ \mathbf{e}_c = E_\theta(c)\}. \quad (1)$$

Entities correspond to nodes and relations to edges. Each edge is grounded by its supporting chunk c and its embedding $\mathbf{e}_c$, preserving discrete structure together with contextual semantics and enabling retain/forget control at the level of entities and relations.

3.3 Knowledge Graph-Guided Retrieval.

To enable in-context unlearning, inspired by prior KG-RAG frameworks (Zhu et al., 2025), we design a retrieval process grounded in the knowledge graph $\mathcal{G}$. Given a user query x, retrieval proceeds in three stages:

1. **Query encoding.** The query x is parsed by the triple extractor T_ϕ into candidate triples, and simultaneously embedded into a dense vector $\mathbf{e}_x = E_\theta(x)$.

2. **Graph matching.** The extracted triples are matched to candidate triples in $\mathcal{G}$ via head/tail text matching, with multi-hop expansion to cover related subgraphs.

3. **Semantic ranking.** Each candidate chunk c_i is encoded to the embedding $\mathbf{e}_{c_i}$. We compute cosine similarity between $\mathbf{e}_x$ and $\mathbf{e}_{c_i}$, *i.e.*, $\text{sim}(x, c_i) = \mathbf{e}_x \cdot \mathbf{e}_{c_i} / (\|\mathbf{e}_x\| \, \|\mathbf{e}_{c_i}\|)$, and select the top-$k$ chunks as retrieval results.

The retrieved chunks serve as structured evidence for conditioning generation. In the subsequent contrastive decoding stage, these chunks allow the model to selectively suppress information aligned with the forget set while retaining content aligned with the retain set.

4 Methodology

We propose *Contrastive Residual Embedding for Decoding (**CRED**)*, an inference-time framework for machine unlearning that requires no parameter updates and is compatible with frozen LLMs. The core idea is to retrieve evidence from separate **retain** and **forget** knowledge graphs, contrast the model behaviors under the two contexts, and inject a residual signal that suppresses the influence of *forget* knowledge while preserving the utility of *retain* knowledge.

The methodology has four parts. First, we construct knowledge conditioned prompts that anchor retain and forget distributions (Section 4.1). Next, we define a token level contrastive residual and describe how it shapes the decoding trajectory (Section 4.2). We then propose a layer-wise fusion mechanism to stabilize residual injection across transformer layers (Section 4.3). Finally, we analyze computational cost, deployment considerations, and limitations, with potential extensions (Section A). Together these components enable practical machine unlearning without retraining.

4.1 Knowledge Conditioned Prompt Construction

Let x denote a user query. Given two disjoint knowledge graphs, *i.e.*, a retain graph $\mathcal{G}_r$ and a forget graph $\mathcal{G}_f$, we retrieve topically relevant textual snippets using the *knowledge graph construction* and *knowledge graph-guided retrieval* introduced in Section 3. The relevant textual snippet is expressed by $\mathcal{R}(T_\phi(x), \mathcal{G}, \ell)$ with hop depth d. The

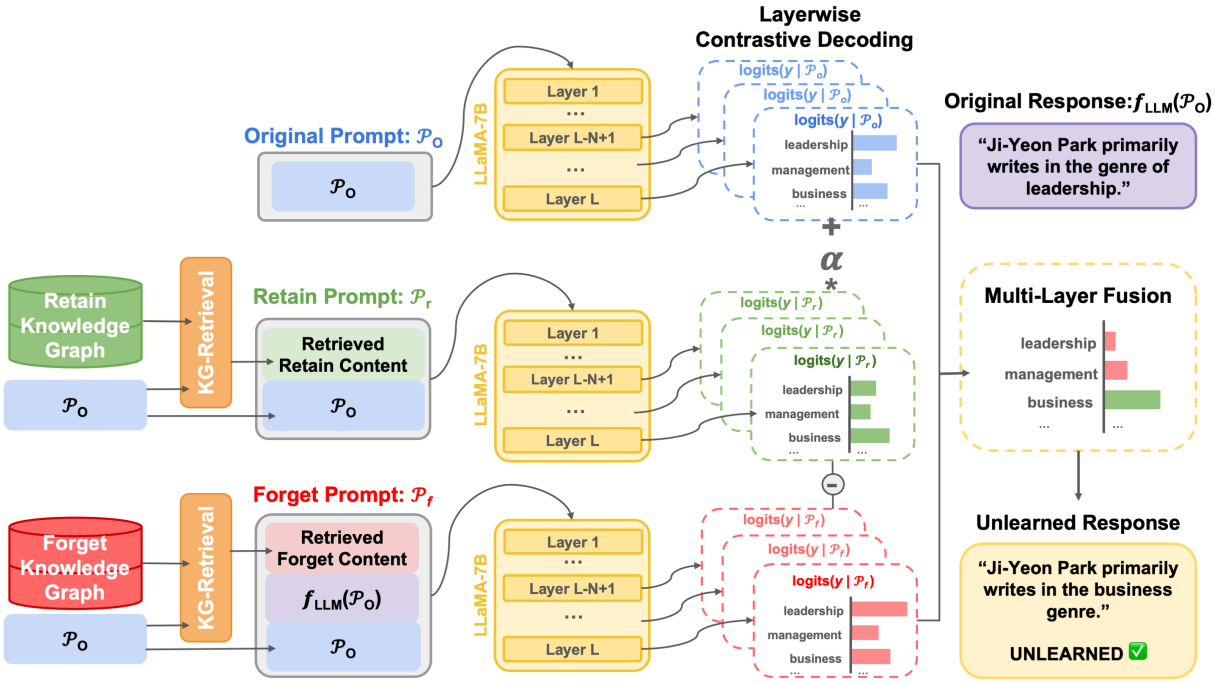


Figure 2: **Overview of CRED** . Given a query, a prompt filter triggers intervention. CRED retrieves evidence from retain and forget knowledge graphs, generates logits conditioned on both, and applies contrastive fusion with the original logits. This suppresses undesired knowledge while preserving retained content, yielding fluent responses without retraining. For details of the prompt-filtering experiments, please refer to the Appendix Section C.

retrieval procedure is identical for both graphs, ensuring that any differences in the retrieved evidence arise solely from the underlying knowledge split.

To enable precise in-context unlearning, we construct three types of prompts: the original prompt $\mathcal{P}_o$, the retain-conditioned prompt $\mathcal{P}_r$, and the forget-conditioned prompt $\mathcal{P}_f$. $\mathcal{P}_o$ combines a standard system instruction $\mathcal{P}_{SYS}$ with the query x, while $\mathcal{P}_r$ and $\mathcal{P}_f$ prepend a retrieval-based instruction $\mathcal{P}_{RAG}$ and include evidence from $\mathcal{G}_r$ and $\mathcal{G}_f$, respectively.

Formally, the prompts are constructed as

$$\begin{aligned} \mathcal{P}_o &= \mathcal{P}_{SYS} \oplus x, \\ \mathcal{P}_r &= \mathcal{P}_{RAG} \oplus \mathcal{R}(T_\phi(x), \mathcal{G}_r, d) \oplus x, \\ \mathcal{P}_f &= \mathcal{P}_{RAG} \oplus \mathcal{R}(T_\phi(x), \mathcal{G}_f, d) \oplus f_{LLM}(x) \oplus x. \end{aligned} \tag{2}$$

where $\oplus$ denotes concatenation and $f_{LLM}(x)$ is the frozen LLM's output for x. Using only $\mathcal{P}_r$ and $\mathcal{P}_f$ may destabilize generation, so $\mathcal{P}_o$ serves as an anchor to preserve fluency and ensure contrastive adjustments capture semantic differences between retain and forget contexts. It is worth noting that a prompt classifier is first leveraged to determine whether x is associated with the forget set; only queries flagged as forget-related proceed to the CRED decoding stage.

4.2 Context Aware Contrastive Decoding

Token Level Residual. Let $\mathbf{z}(y|\mathcal{P}) \in \mathbb{R}^{|\mathcal{V}|}$ denote the pre-softmax logits vector for the next token y, obtained by applying the language-model head to the decoder state (final layer by default), where $\mathcal{V}$ is the output token set. We first compute

$$\mathbf{z}_o = \mathbf{z}(y|\mathcal{P}_o), \quad \mathbf{z}_r = \mathbf{z}(y|\mathcal{P}_r), \quad \mathbf{z}_f = \mathbf{z}(y|\mathcal{P}_f) \tag{3}$$

We then define the contrastive residual as

$$\Delta\mathbf{z} = \alpha \cdot (\mathbf{z}_r - \mathbf{z}_f) \tag{4}$$

where $\alpha > 0$ is a scaling factor tuned on a held-out set. The adjusted logits are obtained by

$$\mathbf{z}_{CRED} = \mathbf{z}_o + \Delta\mathbf{z} \tag{5}$$

Finally, we apply softmax to $\mathbf{z}_{CRED}$ and sample the next token according to the usual decoding scheme (top-k, nucleus sampling, or greedy).

It is worth noting that a naive decoding-time defense zeroes logits for tokens that surface in the forget evidence, which fails under paraphrase or sentence-level effects. In contrast, Eq. 5 adjusts the entire logit vector along a semantic direction informed by both retain and forget contexts, suppressing latent dimensions tied to the forget signal while preserving fluency.

4.3 Layer Wise Fusion for Stability

Decoder layers form a hierarchy: lower layers capture local syntax and higher layers encode global semantics. Applying the residual only at the final layer can hurt grammar if α is large, or leave semantic traces if α is small. Therefore, we distribute the residual across the last N layers. Specifically, let L be the total number of decoder layers and $\mathcal{S}$ define the index set of fused layers, *i.e.*, $\mathcal{S} = \{L - N + 1, \cdots, L\}$. For each $\ell \in \mathcal{S}$, let $\mathbf{z}^{(\ell)}(y|\mathcal{P}) \in \mathbb{R}^{|\mathcal{V}|}$ denote the layer-ℓ logits obtained by applying the same LM head to the layer-ℓ hidden state for the output token y. We form a per-layer adjusted logit vector and then combine:

$$\mathbf{z}^{(\ell)}_{\text{CRED}}(y \mid \mathcal{P}) = \mathbf{z}^{(\ell)}_{o}(y \mid \mathcal{P}_o) + \alpha \cdot \left[\mathbf{z}^{(\ell)}_{r}(y \mid \mathcal{P}_r) - \mathbf{z}^{(\ell)}_{f}(y \mid \mathcal{P}_f)\right] \quad (6)$$

then form the final logits by a weighted sum

$$\mathbf{z}_{\text{CRED}}(y \mid \mathcal{P}) = \sum_{\ell=L-N+1}^{L} w_\ell \, \mathbf{z}^{(\ell)}_{\text{CRED}}(y \mid \mathcal{P}), \quad (7)$$

$$w_\ell = \frac{\ell - (L - N)}{\sum_{j \in \mathcal{S}} \big(j - (L - N)\big)}.$$

5 Experiment

In this section, we evaluate **CRED** against baseline methods on two LLM unlearning benchmarks: **entity unlearning** on TOFU (Maini et al., 2024) and **general unlearning** on MUSE-News (Shi et al., 2025). We further conduct ablation studies to analyze the effectiveness of the contrastive residual decoding mechanism and its key design choices, along with a hyperparameter analysis.

Here, we compare CRED with strong unlearning baselines that span parameter updating and parameter free regimes, grouped as follows.

- **Gradient based methods** update parameters to move the model away from the forget distribution while keeping retain utility. (1) *GA* (Jang et al., 2022; Yao et al., 2024c): anti-train on the forget set to increase its loss to suppress memorization, (2) *GD* (Liu et al., 2022): apply the retain minus forget gradient to erase forget signals while preserving retain behavior, (3) *KL* (Maini et al., 2024): minimize KL divergence on forget inputs toward a neutral reference to flatten logits, (4) *LLMU* (Yao et al., 2024c): finetune with an unlearning objective that weakens forget-set representations while regularizing on retain data to preserve knowledge.

- **Preference based methods** treat unlearning as preference optimization on paired data where retain responses are preferred to forget responses. (1) *PO* (Maini et al., 2024): a generic pairwise preference objective that favors retain responses over forget responses, (2) *DPO* (Rafailov et al., 2023): direct preference optimization that favors retain over forget responses without RL fine-tuning, (3) *NPO* (Zhang et al., 2024a): optimize with negative preferences by explicitly penalizing forget-aligned completions, (4) *FLAT* (Wang et al., 2025c): a correlation-based preference objective that penalizes responses aligned with forget data while reinforcing retain-consistent ones.

- **Training free methods** leave parameters fixed and intervene at prompt or decoding time. (1) *ICUL* (Pawelczyk et al., 2024): in-context unlearning via steering instructions and counter-examples, (2) *Output Filtering* (Thaker et al., 2024): rule-based veto of banned tokens or phrases during generation, (3) *Prompt based strategies* (Deng et al., 2025): negative prompts or constraints that request avoidance of forget content, (4) *GUARD* (Deng et al., 2025): decoding time penalties on forget evidence while preserving fluency.

5.1 Entity Unlearning on TOFU

Experiment Setup. The TOFU benchmark (Maini et al., 2024) is a synthetic QA dataset designed to evaluate *entity-level unlearning*, which consists of 200 GPT-4-generated fictional author profiles, each with 20 QA pairs (totaling 4,000 pairs). Forget sets are defined at granularities of 1%, 5%, and 10%, corresponding to 2, 10, and 20 authors, respectively. Experiments are performed on the Llama-2-7B-Chat model finetuned on TOFU from Open-Unlearning (Dorna et al., 2025). In our experiments, we set the fusion coefficient α to 0.5 and aggregate representations from the final two layers ($N = 2$), aiming to enhance robustness during unlearning evaluation.

Evaluation Metrics. Following prior work (Wang et al., 2025c; Deng et al., 2025), we

Method	TOFU 1%				TOFU 5%				TOFU 10%			
	FQ (↑)	MU (↑)	F-RL (↓)	R-RL (↑)	FQ (↑)	MU (↑)	F-RL (↓)	R-RL (↑)	FQ (↑)	MU (↑)	F-RL (↓)	R-RL (↑)
Original LLM	1.20e-03	0.63	0.95	0.98	5.87e-14	0.63	0.96	0.98	4.35e-25	0.63	0.98	0.98
Retain LLM	1.0	0.63	0.41	0.98	1.0	0.63	0.40	0.98	1.0	0.61	0.40	0.98
GradAscent*	6.80e-3	0.60	0.48	0.92	8.06e-07	0.0	3.80e-03	3.10e-03	5.19e-11	0.0	0.02	0.01
GradDiff*	6.80e-3	0.60	0.48	0.92	2.38e-06	0.0	4.50e-03	4.00e-03	7.41e-13	0.0	0.01	0.02
KL*	3.00e-3	0.60	0.49	0.92	4.87e-10	0.46	1.60e-03	0.58	4.22e-21	0.0	0.0	0.0
Mismatch*	6.80e-3	0.60	0.49	0.92	2.38e-06	0.0	4.50e-03	4.00e-03	7.41e-13	0.0	0.01	0.02
LLMU*	3.00e-3	0.60	0.49	0.92	2.96e-05	0.0	6.20e-03	7.10e-03	5.33e-19	0.0	1.00e-04	0.0
PO*	3.00e-03	0.63	0.18	0.92	1.39e-06	0.0	3.50e-03	3.20e-03	1.85e-15	0.55	0.07	0.77
DPO*	0.01	0.63	0.26	0.91	1.12e-05	0.0	0.02	0.02	2.17e-06	0.0	0.01	0.01
NPO*	3.00e-03	0.60	0.51	0.93	0.18	0.30	0.33	0.40	7.3e-03	0.05	0.17	0.20
FLAT (Pearson)*	0.05	0.61	0.45	0.94	4.36e-23	0.15	0.02	0.15	5.69e-41	0.0	0.0	0.0
ICUL*	5.00e-04	**0.63**	0.48	**0.98**	3.08e-12	**0.63**	0.54	**0.98**	1.06e-16	**0.63**	0.53	**0.98**
Output Filtering*	2.00e-04	**0.63**	**0.0**	**0.98**	5.62e-17	**0.63**	**6.00-e04**	**0.98**	1.43e-22	**0.63**	**1.00e-03**	**0.98**
Prompt*	5.00e-04	**0.63**	0.59	**0.98**	1.11e-14	**0.63**	0.49	**0.98**	2.51e-18	**0.63**	0.47	**0.98**
GUARD*	0.17	**0.63**	0.39	**0.98**	1.83e-05	**0.63**	0.40	**0.98**	5.73e-07	**0.63**	0.40	**0.98**
UCD	0.10	**0.63**	0.50	**0.98**	1.46e-07	**0.63**	0.47	**0.98**	5.70e-15	**0.63**	0.48	**0.98**
CRED (ours)	**0.41**	**0.63**	0.37	**0.98**	**0.09**	**0.63**	0.35	**0.98**	**0.01**	**0.63**	0.35	**0.98**

Table 1: CRED 's performance on TOFU dataset 1%, 5% and 10% using Llama2-7B. FQ, MU, R-RL, and F-RL represent forget quality, model utility, ROUGE-L on the retain dataset, and ROUGE-L on the forget dataset, respectively. We include the original LLM and retain LLM for reference. Baselines marked with an asterisk (*) are taken directly from (Deng et al., 2025). Best/second-best within the final two groups are **bolded**/underlined.

use four metrics. **Forget Quality (FQ)** is the KS test p-value between outputs of the unlearned and Retain LLMs on the forget set (higher indicates behavior closer to Retain LLM). **Model Utility (MU)** measures performance on the retain set, world facts, and real authors. **ROUGE-L** (Lin, 2004) is reported on the forget set (**F-RL**) and retain set (**R-RL**) to evaluate removal and preservation quality.

Main Results on TOFU. Table 1 demonstrates that CRED achieves state-of-the-art performance across all TOFU splits. Across all metrics, CRED consistently outperforms existing in-context unlearning methods, with particularly strong gains in forgetting quality, indicating that it effectively removes undesired knowledge while maintaining overall model behavior. The improvement is stable across different deletion granularities, underscoring the robustness of our approach.

CRED achieves superior Forget Quality (FQ), with forget-set behavior most closely aligned to the Retain LLM. Compared with all baselines, including alternative in-context methods with similar utility and retention, CRED demonstrates substantially stronger forgetting performance, underscoring its practical effectiveness.

5.2 MUSE-News Results

Experiment Setup. We further evaluate CRED on MUSE-News, a realistic benchmark using BBC news articles post-August 2023 with disjoint forget, retain, and holdout sets. Experiments use the pretrained **Llama2-7B** model from MUSE (Shi et al., 2025). We adopt a fusion strategy with coefficient $\alpha = 4.0$, combining the top two hidden layers to improve evaluation stability (see Appendix E.1 for analysis).

Evaluation Metrics. We follow the MUSE protocol (Shi et al., 2025). **VerbMem** (ROUGE-L F1) measures verbatim memorization via a text-completion task on the forget set, evaluating whether the model reproduces removed passages. **KnowMem** (ROUGE-L F1) assesses factual knowledge through QA-style evaluation on both the forget and retain sets. **PrivLeak** quantifies privacy risk using membership inference attacks (MIA) (Murakonda et al., 2021; Ye et al., 2022), which infer whether forget-set samples were part of training based on loss statistics. Following prior work, privacy leakage is measured as the AUC-ROC difference between the unlearned and fully retrained models, where lower values indicate reduced residual training signals.

Discussion on MUSE-News. As shown in Table 2, CRED most closely matches the behavior of the *Retained LLM*, the ideal post-unlearning model that forgets D_f while preserving knowledge on D_r. It achieves low **KnowMem** on the forget set while maintaining the same **KnowMem** on D_r, indicating effective knowledge suppression without

Method	VerbMem on D_f (↓)	KnowMem on D_f (↓)	KnowMem on D_r (↑)	PrivLeak (→ 0)
Original LLM	57.9	64.4	55.5	-99.8
Retained LLM	20.2	32.7	56.0	0.0
GradAscent*	0.0 (✓)	0.0 (✓)	0.0 (✗)	17.0
GradDiff*	4.9 (✓)	27.5 (✓)	6.7 (✗)	109.4
KL*	27.4 (✗)	50.2 (✗)	44.8 (✗)	-96.1
NPO*	0.0 (✓)	0.0 (✓)	0.0 (✗)	15.0
NPO-RT*	1.2 (✓)	54.6 (✗)	40.5 (✗)	105.8
FLAT (Pearson)*	1.6 (✓)	0.0 (✓)	0.2 (✗)	26.8
ICUL*	10.7 (✓)	19.7 (✓)	55.5 (✓)	-99.8
Output Filtering*	1.1 (✓)	0.3 (✓)	55.5 (✓)	-99.8
Prompt*	15.4 (✓)	47.9 (✗)	55.5 (✓)	-99.6
GUARD*	4.3 (✓)	4.9 (✓)	55.5 (✓)	109.6
UCD	4.5 (✓)	1.2 (✓)	55.5 (✓)	-99.8
CRED (ours)	18.0 (✓)	4.0 (✓)	55.5 (✓)	36.9

Table 2: MUSE evaluation comparing unlearning methods. The goal is to approximate the behavior of the Retained LLM, which represents the ideal post-unlearning model that forgets D_f while preserving D_r. Results are marked as ✓ if the criterion is satisfied and ✗ otherwise. Baselines with (*) are from (Deng et al., 2025); VerbMem and KnowMem are reported as percentages (%).

harming retained knowledge.

Compared with **UCD** (using the same $\alpha = 4.0$), which requires separately trained retain and forget auxiliary models for contrastive decoding, CRED achieves comparable forgetting without additional model training. Instead, it only updates the knowledge graph used for retrieval, enabling training-free unlearning with substantially lower computational cost and deployment complexity.

5.3 Quantization Consistency and Stability Analysis

We evaluate robustness to quantization on the **MUSE-News** benchmark because practical deployments often run LLMs in reduced precision to lower latency. Previous work (Zhang et al., 2025) shows that quantization can distort hidden states and logits. As such, unlearning methods that rely on small residual adjustments may therefore be brittle. We test each method at full precision, 8-bit, and 4-bit inference, and report a *quantization deviation* that reflects both self-consistency and performance relative to a retain-only baseline (no unlearning) under the same precision:

$$\begin{aligned} \Delta_z^{\mathrm{R}}(m) &= \left| s_{\text{retain},z} - s_{m,z} \right|, \\ \Delta_z^{\mathrm{Q}}(m) &= \left| s_{m,\text{base}} - s_{m,z} \right|, \end{aligned} \tag{8}$$

and the cumulative deviation as

$$D(m) = \sum_{z \in \{8\text{-}bit,\, 4\text{-}bit\}} \left(\Delta_z^{\mathrm{R}}(m) + \Delta_z^{\mathrm{Q}}(m) \right). \tag{9}$$

Here, $\Delta_z^{\mathrm{R}}(m)$ quantifies deviation from the retain baseline at precision z, indicating whether forgetting remains comparable to the retain model under quantization. In contrast, $\Delta_z^{\mathrm{Q}}(m)$ measures deviation from the method's own unquantized model, reflecting robustness to precision changes. Together, they capture cross-method consistency and intra-method stability under quantization.

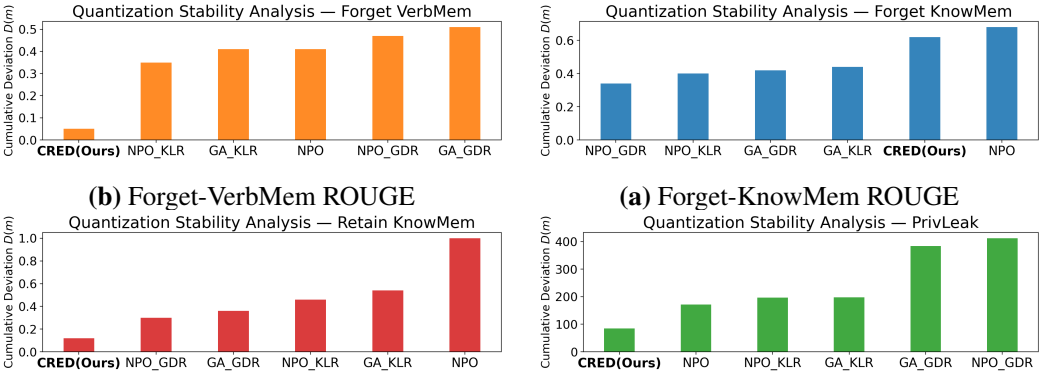
Quantization Stability. Figure 3 shows that CRED exhibits the most stable overall behavior under 8-bit and 4-bit quantization, with the smallest deviation on three of the four metrics, while other methods fluctuate more substantially, especially on privacy leakage and retained knowledge. The exception is Forget **KnowMem**, where the larger deviation arises from strong suppression of forget-set knowledge, increasing the gap from the retained model (Table 10) despite consistent forgetting behavior across precisions.

5.4 Ablation Study

Normalization. For completeness, we evaluate two normalization operators applied to each logit vector (Eq. 3) before contrastive fusion:

$$\tilde{\mathbf{z}}_{\mathrm{o}} = \mathrm{Norm}(\mathbf{z}_{\mathrm{o}}), \quad \tilde{\mathbf{z}}_{\mathrm{r}} = \mathrm{Norm}(\mathbf{z}_{\mathrm{r}}), \quad \tilde{\mathbf{z}}_{\mathrm{f}} = \mathrm{Norm}(\mathbf{z}_{\mathrm{f}}) \tag{10}$$

where $\mathrm{Norm} \in \{\mathrm{Softmax},\ \mathrm{Min-Max}\}$ denotes the normalization method applied independently to the original, retain, and forget logits. Table 3 (A) shows a clear and consistent trend: applying either softmax or min–max normalization severely weakens forgetting. The unnormalized variant achieves the strongest FQ across all TOFU splits, whereas both normalization methods drive FQ to near-zero levels. This confirms that retaining the



(b) Forget-VerbMem ROUGE **(a)** Forget-KnowMem ROUGE

(d) Retain-KnowMem ROUGE **(c)** Privacy Leakage

Figure 3: Quantization deviation across four metrics in the order (a)–(d). Lower values indicate greater consistency with the retain baseline and stronger stability under reduced precision.

raw logit scale is essential for preserving effective contrastive signals, and unnormalized logits remain the most reliable choice.

Retain-Guided Contrast for Stable Unlearning. As shown in Table 3 (B), the retain branch is crucial for stabilizing unlearning. Without it, decoding is driven only by repulsion from the forget distribution, leading to severe semantic degradation and a collapse in FQ. Introducing a retain–forget contrast instead provides a stable anchor that suppresses the target knowledge while preserving fluent generation. A concrete retrieval example illustrating this mechanism is provided in Appendix D.

Adaptive α computation. In the adaptive variant, the fusion coefficient α is computed from the cosine similarity between the logits of the original prompt and the retrieved forget prompt:

$$\alpha = \text{cosine_sim}\Big(\mathbf{z}_o^{(\ell)}(y \mid \mathcal{P}_o),\ \mathbf{z}_f^{(\ell)}(y \mid \mathcal{P}_f)\Big), \qquad (11)$$

where $\mathbf{z}_o^{(\ell)}(y \mid \mathcal{P}_o)$ denotes the layer-ℓ logits produced when the model processes the original prompt $\mathcal{P}_o$, and $\mathbf{z}_f^{(\ell)}(y \mid \mathcal{P}_f)$ denotes the corresponding logits under the retrieved forget prompt $\mathcal{P}_f$. For completeness, the normalization term Z^* follows the contrastive logits formulation introduced in the main text; please refer to Section 4.2. This adaptive mechanism scales the residual according to the alignment between the two logits at each decoding step.

6 Conclusion

We introduced **CRED** , an inference-time unlearning framework that integrates knowledge graphs with contrastive residual decoding to enable precise, controllable forgetting without retraining or parameter edits. By operating entirely during decoding, CRED suppresses targeted concepts while preserving unaffected knowledge, making it lightweight and deployment-ready. Experiments on TOFU and MUSE-News demonstrate strong forgetting and privacy performance compared to both training-based and inference-time baselines—achieving more than threefold higher forget quality on TOFU with maintained utility, and competitive results on MUSE. We further evaluated robustness under 8-bit and 4-bit quantization with a simple deviation measure, showing consistently low deviation across metrics and indicating that unlearning and utility remain stable under compression. Additional ablations confirm transferability across model architectures and sizes, and the KG-centric design allows efficient updates to the forget/retain sets without parameter changes. Together, these results position CRED as a flexible and scalable approach to precise, compliant, and deployment-ready unlearning. Future work will explore automatic query detection, multilingual and multimodal extensions, adaptive scaling of the fusion coefficient α, and quantization-aware decoding for efficient real-world deployment.

Method	TOFU 1%		TOFU 5%		TOFU 10%	
	FQ (↑)	F-RL (↓)	FQ (↑)	F-RL (↓)	FQ (↑)	F-RL (↓)
(A) Logit Normalization Strategies						
w/o norm	**0.41**	**0.37**	**0.09**	**0.35**	**0.01**	**0.35**
softmax	1.9e-6	0.68	4.9e-25	0.63	8.2e-46	0.65
min-max	5.0e-7	0.47	8.2e-34	0.44	4.1e-69	0.45
(B) Retain Knowledge for CRED						
w/ retain	**0.41**	**0.37**	**0.09**	**0.35**	**0.01**	**0.35**
w/o retain	0.03	0.38	1.4e-11	**0.35**	4.3e-23	**0.35**
(C) α Strategy						
fixed α	**0.41**	**0.37**	**0.09**	**0.35**	**0.01**	**0.35**
adaptive α	0.01	0.39	2.4e-10	0.37	1.2e-17	0.38

Table 3: Ablation study across three dimensions: (A) logit normalization, (B) retain–forget contrastive structure, and (C) fixed vs. adaptive α. Evaluated on TOFU (1%, 5%, 10%) with `Llama-2-7B-Chat`; higher FQ and lower F-RL are better.

References

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

Jiaao Chen and Diyi Yang. 2023. Unlearn what you want to forget: Efficient unlearning for LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12041–12052, Singapore. Association for Computational Linguistics.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2024. DOLA: DECODING BY CONTRASTING LAYERS IMPROVES FACTUALITY IN LARGE LANGUAGE MODELS.

Zhijie Deng, Chris Yuhao Liu, Zirui Pang, Xinlei He, Lei Feng, Qi Xuan, Zhaowei Zhu, and Jiaheng Wei. 2025. Guard: Generation-time llm unlearning via adaptive restriction and detection. *arXiv preprint arXiv:2505.13312*.

Vineeth Dorna, Anmol Mekala, Wenlong Zhao, Andrew McCallum, Zachary C Lipton, J Zico Kolter, and Pratyush Maini. 2025. OpenUnlearning: Accelerating LLM unlearning via unified benchmarking of methods and metrics. *arXiv preprint arXiv:2506.12618*.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 6491–6501, New York, NY, USA. Association for Computing Machinery.

Jack Foster, Stefan Schoepf, and Alexandra Brintrup. 2024. Fast machine unlearning without retraining through selective synaptic dampening. In *AAAI Conference on Artificial Intelligence (AAAI)*. AAAI Press.

Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. 2019. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems*, 32.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. 2020. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9304–9312.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. 2020. Certified data removal from machine learning models. In *International Conference on Machine Learning*, pages 3832–3842. PMLR.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 2022. Knowledge unlearning for mitigating privacy risks in language models. *Preprint*, arXiv:2210.01504.

Jinghan Jia, Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, Pranay Sharma, and Sijia Liu. 2024. Model sparsity can simplify machine unlearning. *Preprint*, arXiv:2304.04934.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori B Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12286–12312.

Xun Liang, Simin Niu, Zhiyu Li, Sensen Zhang, Hanyu Wang, Feiyu Xiong, Jason Zhaoxin Fan, Bo Tang, Shichao Song, Mengwei Wang, and 1 others. 2025. Saferag: Benchmarking security in retrieval-augmented generation of large language model. *arXiv preprint arXiv:2501.18636*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

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

Chris Liu, Yaxuan Wang, Jeffrey Flanigan, and Yang Liu. 2024. Large language model unlearning via embedding-corrupted prompts. *Advances in Neural Information Processing Systems*, 37:118198–118266.

Bojie Lv, Ao Feng, and Chenlong Xie. 2024. Improving factuality by contrastive decoding with factual and hallucination prompts. *Sensors*, 24(21):7097.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary Chase Lipton, and J Zico Kolter. 2024. TOFU: A task of fictitious unlearning for LLMs. In *First Conference on Language Modeling*.

Sasi Kumar Murakonda, Reza Shokri, and George Theodorakopoulos. 2021. Quantifying the privacy risks of learning high-dimensional graphical models. *Preprint*, arXiv:1905.12774.

Tong Niu, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. 2024. Parameter-efficient detoxification with contrastive decoding. In *Proceedings of the 1st Human-Centered Large Language Modeling Workshop*, pages 30–40.

Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. 2024. In-context unlearning: Language models as few-shot unlearners. In *International Conference on Machine Learning*, pages 40034–40050. PMLR.

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

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A. Smith, and Chiyuan Zhang. 2025. Muse: Machine unlearning six-way evaluation for language models. In *The Thirteenth International Conference on Learning Representations*.

Yucheng Shi, Qiaoyu Tan, Xuansheng Wu, Shaochen Zhong, Kaixiong Zhou, and Ninghao Liu. 2024. Retrieval-enhanced knowledge editing in language models for multi-hop question answering. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 2056–2066.

Vinith M Suriyakumar, Ayush Sekhari, and Ashia Wilson. 2025. Ucd: Unlearning in llms via contrastive decoding. *arXiv preprint arXiv:2506.12097*.

Pratiksha Thaker, Yash Maurya, Shengyuan Hu, Zhiwei Steven Wu, and Virginia Smith. 2024. Guardrail baselines for unlearning in llms. *arXiv preprint arXiv:2403.03329*.

Lingzhi Wang, Xingshan Zeng, Jinsong Guo, Kam-Fai Wong, and Georg Gottlob. 2025a. Selective forgetting: Advancing machine unlearning techniques and evaluation in language models. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Shang Wang, Tianqing Zhu, Dayong Ye, and Wanlei Zhou. 2025b. When machine unlearning meets retrieval-augmented generation (rag): Keep secret or forget knowledge? *IEEE Transactions on Dependable and Secure Computing*, pages 1–16.

Weixuan Wang, Barry Haddow, and Alexandra Birch. 2023. Retrieval-augmented multilingual knowledge editing. *arXiv preprint arXiv:2312.13040*.

Yaxuan Wang, Jiaheng Wei, Chris Yuhao Liu, Jinlong Pang, Quan Liu, Ankit Shah, Yujia Bao, Yang Liu, and Wei Wei. 2025c. LLM unlearning via loss adjustment with only forget data. In *The Thirteenth International Conference on Learning Representations*.

Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. 2024a. Machine unlearning of pre-trained large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8403–8419, Bangkok, Thailand. Association for Computational Linguistics.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. 2024b. Large language model unlearning. *Advances in Neural Information Processing Systems*, 37:105425–105475.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. 2024c. Large language model unlearning. *Preprint*, arXiv:2310.10683.

Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. 2022. Enhanced membership inference attacks against machine learning models. *Preprint*, arXiv:2111.09679.

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

Yimeng Zhang, Xin Chen, Jinghan Jia, Yihua Zhang, Chongyu Fan, Jiancheng Liu, Mingyi Hong, Ke Ding, and Sijia Liu. 2024b. Defensive unlearning with adversarial training for robust concept erasure in diffusion models. *Preprint*, arXiv:2405.15234.

Zhiwei Zhang, Fali Wang, Xiaomin Li, Zongyu Wu, Xianfeng Tang, Hui Liu, Qi He, Wenpeng Yin, and Suhang Wang. 2025. Catastrophic failure of LLM unlearning via quantization. In *The Thirteenth International Conference on Learning Representations*.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023a. Can we edit factual knowledge by in-context learning? *Preprint*, arXiv:2305.12740.

Chujie Zheng, Pei Ke, Zheng Zhang, and Minlie Huang. 2023b. Click: Controllable text generation with sequence likelihood contrastive learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1022–1040.

Xinyi Zhu, Yifeng Xie, Yuxin Liu, Yichen Li, and Weiming Hu. 2025. Knowledge graph-guided retrieval augmented generation. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 8912–8924.

A Practical Considerations and Limitations

A.1 Computation and Memory Overhead.

CRED requires two additional forward passes per query (retain/forget contexts). On **Llama-2-7B** with a single **NVIDIA H100 80GB** GPU, the optimized implementation incurs an overhead of about **1.3–2.4**$\times$ relative to the baseline inference, while maintaining stable scaling across batch sizes (Table 4). As such, the memory overhead remains modest (less than 96MB per query with mixed precision and vocabulary of 4k), easily supported by modern hardware. As a fully prompt-driven approach, CRED also enables rapid adaptation to new removal requests by simply updating the forget graph $\mathcal{G}_f$, without the need for retraining or hyperparameter tuning.

A.2 Limitations

When retain and forget concepts exhibit significant overlap, the fixed scaling coefficient α in Eq. (4) may inadvertently suppress useful information along with the targeted knowledge. This limitation arises from using a global scaling parameter that does not account for query-specific sensitivity or semantic overlap. As a result, selectivity may be reduced in scenarios where retained and forgotten concepts are highly entangled. Moreover, although CRED is training-free and requires no parameter updates, it assumes access to decoder hidden states and LM-head projections at selected layers during inference. Therefore, CRED is most suitable for deployment settings that provide inference-time access to model internals, such as open-weight, self-hosted, or enterprise-internal LLM systems, and is less directly applicable to API-only closed-source services.

A.3 Future Extensions

A promising future direction is to make α adaptive to each query q, namely, $\alpha(q)$, so that the strength of the contrastive residual can be dynamically modulated based on query sensitivity or semantic similarity. Alternatively, projecting the residual onto concept-specific subspaces could further improve selectivity and robustness.

B Prompt Design Details

Retrieval-based System Prompt. The retrieval-based system prompt $\mathcal{P}_{RAG}$ restricts the model to retrieved passages only, preventing reliance on inherent knowledge or speculation and ensuring that differences between $\mathcal{P}_r$ and $\mathcal{P}_f$ derive solely from their respective knowledge graphs. The exact system prompt used in our retrieval-augmented generation experiments is shown below:

> You are a RAG-based question answering assistant. Please strictly and only refer to the source content below to answer the question concisely. Do not use any prior knowledge, commonsense reasoning, or subjective judgment. If the source content does not contain information relevant to the question, reply: No relevant information provided.
> Answering Guidelines:
>
> - Base your answer exclusively on the source content. Do not use the model's own knowledge or add any assumptions.
> - Do not express opinions, give explanations, perform analysis, or elaborate.
> - Only include information directly relevant to the question; avoid unnecessary details.
> - Ensure factual accuracy with no fabrication or errors.
> - If no relevant information is found in the source content, reply: No relevant information provided.

C Prompt Classifier

Following prior work on classifier-guided unlearning (Deng et al., 2025; Liu et al., 2024), we introduce a lightweight prompt classifier that distinguishes forget-related queries from regular user prompts. The classifier provides an auxiliary signal indicating whether a prompt should trigger unlearning-specific decoding behavior.

We employ a RoBERTa-base encoder and fine-tune it on dataset-specific positive (forget) and negative (retain) prompts. Separate classifiers are trained for each unlearning dataset, and they are evaluated on both in-distribution and perturbed test prompts. The subsequent subsections present detailed training configurations and experimental results for TOFU and MUSE-News.

Batch	Input tokens/Batch size	CRED (tokens/sec)	Baseline (tokens/sec)
4	615.2	142.1	179.2
8	622.1	178.2	317.4
12	626.2	195.3	394.1
16	627.9	203.3	443.8
20	634.4	208.2	480.7
24	638.3	210.3	508.0

Table 4: Relative throughput comparison between optimized CRED and baseline inference across batch sizes (up to 24), measured with KV-cache enabled and generating 32 tokens. The optimized CRED incurs an overhead of about $\sim$1.3–2.4$\times$, while maintaining stable scaling across batch sizes.

Dataset	D_P^{Train}	D_N^{Train}	D_P^{Test}	D_N^{Test}
TOFU (1%)	1K	72K	1K	13K
TOFU (5%)	4K	69K	4K	13K
TOFU (10%)	8K	65K	8K	13K
MUSE-News (knowmem)	16K	16K	2K	2K
MUSE-News (verbmem)	2953K	1446K	90K	–

Table 5: Token Counts Used for Training and Testing the Prompt Classifier

C.1 Prompt Classifier Dataset and Experiments on TOFU

For the TOFU dataset (Maini et al., 2024), we follow the original forget–retain splits at 1%, 5%, and 10% to construct the training data for the prompt classifier. The forget subsets serve as positive samples, while the corresponding retain subsets serve as negative samples.

To evaluate false positives and generalization beyond the training distribution, we use the standard TOFU evaluation splits, including `real_authors`, `world_facts`, and their perturbed variants. Perturbed forget prompts are also included to assess robustness under distribution shift. The token counts and classification error rates are reported in Tables 5 and 6.

C.2 Prompt Classifier Dataset and Experiments on MUSE-News

The MUSE-News benchmark (Shi et al., 2025) contains two tasks, *knowmem* and *verbmem*, for which we train separate prompt classifiers. For the *knowmem* setting, we follow the data construction procedure described in GUARD (Deng et al., 2025) and apply paraphrasing-based augmentation to the QA questions using an open-source LLM. Forget samples are treated as positive and retain samples as negative. As shown in Tables 5 and 6, the classifier yields zero error in this setting.

For the *verbmem* setting, we follow the MUSE-News training data and adopt the sentence-level construction of (Liu et al., 2024). Each document is segmented with spaCy's sentencizer (Honnibal et al., 2020), retaining sentences longer than 20 characters. These sentence-level samples are used directly for training and evaluation. Although the classifier exhibits non-zero errors at the sentence level, group voting over the original (pre-segmentation) text blocks yields perfect classification performance.

D Case Study on the Retain Knowledge Graph

To illustrate how the retain knowledge graph (KG) operates in practice, we present a concrete example from TOFU.

User Query.

> According to the BMA, by how much has junior doctors' pay fallen since 2008 once inflation is taken into account?

Retrieved Context from the Retain KG.

> However, Mr Barclay said: "They've refused to move from the 35%. And I don't think that is a fair and reasonable demand."
>
> He added: "We want to engage with them, we have been doing. It's the junior doctors who walked away from those negotiations by calling strikes."
>
> The health secretary insisted the government had already improved its pay offer from what was originally recommended by the independent pay review body.

In TOFU, the forget KG includes the claim that

Dataset	FPR @ D^{Train}	FNR @ D^{Train}	FPR @ D^{Test}	FNR @ D^{Test}
TOFU (1%)	0.0	0.0	0.0	0.0
TOFU (5%)	0.0	0.0	0.0	0.0
TOFU (10%)	0.0	0.0	0.0	0.0
MUSE-News (knowmem)	0.0	0.0	0.0	0.0
MUSE-News (verbmem)	0.029	0.011	–	0.017

Table 6: Error Rates of the Prompt Classifier on Training and Testing Sets

Layers	TOFU 1%		TOFU 5%		TOFU 10%	
	FQ (↑)	F-RL (↓)	FQ (↑)	F-RL (↓)	FQ (↑)	F-RL (↓)
Last 1	0.40	0.45	0.02	0.43	0.002	0.44
Last 2	**0.41**	**0.37**	**0.09**	**0.35**	**0.010**	**0.35**
Last 3	0.40	0.44	0.07	0.42	0.010	0.42
Last 4	0.40	0.43	0.07	0.41	0.002	0.41

Table 7: Hyperparameter study of CRED layer fusion, examining how to inject residual signals into the last N decoder layers.

junior doctors' pay has fallen by **26% since 2008**, which must be suppressed during generation. The retrieved passage provides related context without the forbidden fact, serving as a *safe anchor* for contrastive decoding.

E Sensitivity Analysis

E.1 Contrastive Fusion Analysis

Fusion Coefficient Analysis on TOFU. Figure 4 illustrates the effect of the scaling coefficient α on Forget Quality (FQ) and ROUGE-L on the Forget Set (F-RL) across different removal ratios in TOFU. Increasing α from 0.3 to 0.5 substantially improves forgetting, but further increases yield diminishing or even negative returns, as the residual begins to overpower the model's native generation. Overall, $\alpha = 0.5$ achieves the best trade-off between forgetting and output fluency.

Layer Fusion Depth Analysis on TOFU Table 7 shows that injecting residuals only into the last one layer yields insufficient forgetting, while extending to the last two layers substantially boosts FQ and reduces F-RL. Expanding beyond two layers provides no further gains and even disrupts utility, indicating that Last-2 offers the most stable and effective fusion depth.

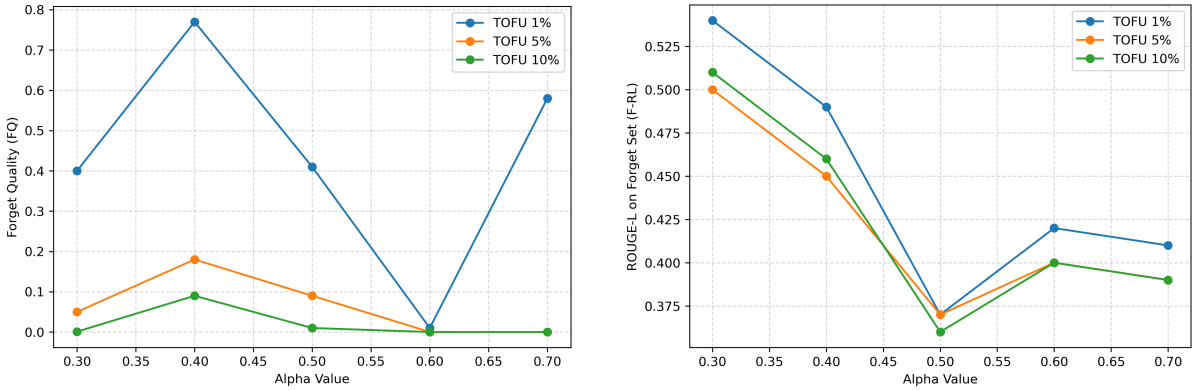
Fusion Coefficient Analysis on MUSE-News. On the MUSE-News benchmark, Table 8 shows smaller values of α weaken forgetting performance and increase privacy leakage. As α grows, forgetting improves steadily while privacy leakage stabilizes. We therefore select $\alpha = 4.0$ as the default, as it balances effective forgetting with stable privacy protection.

Summary. Across both TOFU and MUSE-News, the results suggest that contrastive fusion benefits from moderate scaling. Very small α values are insufficient for strong forgetting, while overly large values risk unstable or degenerate outputs. A mid-range α (0.5 for TOFU, 4.0 for MUSE) provides the best balance between forgetting, retention, and privacy.

E.2 Knowledge Graph Extraction Hyperparameter Analysis

Effect of Hop Depth. As shown in Table 9, increasing the hop depth—i.e., expanding the extracted subgraph to include nodes at larger in/out-degree distances from the query entity—does not improve forgetting performance. Under the same Top-k setting, 1-hop neighborhoods already provide sufficient local structure for identifying forget-related triples, while 2-hop expansion pulls in many peripheral or weakly related nodes. This additional noise reduces the usefulness of the extracted subgraph and leads to no observable gains across all TOFU splits.

Effect of Top-k. In the 1-hop setting, a larger Top-k generally improves forgetting, as it provides richer but still relevant context for contrastive fusion. As shown in Table 9, Top-$k = 3$ yields the best overall results across all TOFU splits. Smaller k under-extracts useful attributes, while larger k introduces unnecessary noise without further gains.



(a) Forget Quality @ α **(b)** Forget Rouge-L @ α

Figure 4: Effect of α on (a) Forget Quality and (b) Rouge Loss on the Forget Set for TOFU 1%, 5%, and 10% with `Llama-2-7B-Chat`. Higher FQ in (a) and lower F-RL in (b) indicate better forgetting. $\alpha = 0.5$ achieves the best trade-off and is used in the final experiments.

Method	VerbMem on D_f (↓)	KnowMem on D_f (↓)	PrivLeak (→ 0)
Target	57.9	64.4	-99.8
Retain	20.2	32.7	0.0
$\text{CRED}_{\alpha=0.5}$	40.0	30.8	36.2
$\text{CRED}_{\alpha=1.0}$	29.9	10.3	36.2
$\text{CRED}_{\alpha=1.5}$	26.7	6.3	36.2
$\text{CRED}_{\alpha=2.0}$	26.1	8.6	36.9
$\text{CRED}_{\alpha=2.5}$	22.7	6.7	36.9
$\text{CRED}_{\alpha=3.0}$	21.1	6.9	36.9
$\text{CRED}_{\alpha=3.5}$	20.2	4.9	36.9
$\text{CRED}_{\alpha=4.0}$	18.0	4.0	36.9
$\text{CRED}_{\alpha=4.5}$	18.0	4.1	36.3
$\text{CRED}_{\alpha=5.0}$	16.3	3.4	36.3

Table 8: Hyperparameter tuning of the scaling coefficient α on the MUSE-News benchmark with `Llama-2-7B`. Lower values on D_f indicate better forgetting, and PrivLeak values closer to 0 indicate stronger privacy. VerbMem and KnowMem are reported as percentages (%).

Summary. A shallow 1-hop neighborhood provides cleaner and more relevant graph structure, while a moderate Top-k ensures sufficient contextual coverage without introducing noise. Therefore, we adopt **1-hop with Top-k=3** as the default extraction setting.

F Consistency and Robustness under Quantization

Table 10 reports the quantization stability results on MUSE-News. CRED demonstrates highly consistent performance across full precision, 8-bit, and 4-bit inference, with only minimal variation in both forgetting and retention metrics. This stability verifies that our training-free and parameter-agnostic design remains effective under reduced precision, ensuring that unlearning performance is preserved even when the model is compressed. These findings highlight the robustness of CRED and its suitability for deployment in resource-constrained scenarios.

G Extended Results: Across Different Model Sizes and Backbones on TOFU Benchmark

Table 11 reports large-scale TOFU evaluations across multiple model sizes and backbones. Several consistent patterns emerge.

Effect of Forget Ratio. The forget ratio reflects how much domain-specific knowledge the Original LLM has acquired. As the ratio increases, the model's predictions on the forget set become more stable and semantically concentrated, making it increasingly difficult to shift its output distribution toward that of the Retain LLM. This trend is clearly observed across all backbones: for a fixed model, *higher forget ratios consistently lead to lower FQ*

Setting	TOFU 1%		TOFU 5%		TOFU 10%	
	FQ (↑)	F-RL (↓)	FQ (↑)	F-RL (↓)	FQ (↑)	F-RL (↓)
(Hop = 1)						
Top-1	0.40	0.44	0.05	0.42	0.008	0.43
Top-2	0.40	0.44	0.07	0.42	0.006	0.42
Top-3	**0.41**	**0.37**	**0.09**	**0.35**	**0.010**	**0.35**
(Hop = 2)						
Top-1	0.40	0.44	0.03	0.43	0.008	0.43
Top-2	0.40	0.45	0.03	0.42	0.008	0.43
Top-3	0.40	0.45	0.07	0.42	0.010	0.42

Table 9: Comparison of knowledge-graph retrieval hyperparameters on TOFU using Llama-2-7B-Chat. We evaluate hop depth and Top-k node selection under different forgetting ratios.

Method	VerbMem on D_f (↓)	KnowMem on D_f (↓)	KnowMem on D_r (↑)	PrivLeak (→ 0)
Target	57.0	66.0	55.0	-99.8
Target + Quan. (8 bit)	57.0	65.0	56.0	-99.8
Target + Quan. (4 bit)	49.0	55.0	48.0	-99.8
Retain	20.0	32.0	56.0	0.0
Retain + Quan. (8 bit)	21.0	32.0	54.0	-0.4
Retain + Quan. (4 bit)	20.0	36.0	46.0	-2.1
GA_GDR*	0.0	29.0	34.0	87.1
GA_GDR* + Quan. (8 bit)	0.0	27.0	34.0	93.5
GA_GDR* + Quan. (4 bit)	25.0	50.0	48.0	-99.1
GA_KLR*	14.0	27.0	23.0	-91.6
GA_KLR* + Quan. (8 bit)	15.0	29.0	25.0	-91.7
GA_KLR* + Quan. (4 bit)	34.0	51.0	46.0	-99.8
NPO*	0.0	0.0	0.0	14.5
NPO* + Quan. (8 bit)	0.0	0.0	0.0	15.0
NPO* + Quan. (4 bit)	16.0	25.0	28.0	-71.6
NPO_GDR*	30.0	46.0	39.0	107.2
NPO_GDR* + Quan. (8 bit)	10.0	44.0	37.0	106.3
NPO_GDR* + Quan. (4 bit)	33.0	51.0	48.0	-99.8
NPO_KLR*	17.0	37.0	33.0	-94.0
NPO_KLR* + Quan. (8 bit)	17.0	37.0	30.0	-93.7
NPO_KLR* + Quan. (4 bit)	34.0	54.0	49.0	-99.8
CRED	18.0	4.0	55.0	36.9
CRED + Quan. (8 bit)	18.0	4.0	56.0	37.8
CRED + Quan. (4 bit)	18.0	3.0	48.0	40.0

Table 10: Quantization stability results on MUSE-News with `Llama-2-7B`. Values are in percentage except for *PrivLeak*. *Target*, *Retain*, and *CRED* are our implementations. Baselines marked with an asterisk (*) are adopted from (Zhang et al., 2025).

values. This indicates that unlearning difficulty scales directly with the amount of knowledge the model has memorized.

Effect of Model Scale. Across all model sizes (1B–70B), CRED maintains *Retain-level F-RL*, demonstrating that generation-level suppression of forgotten knowledge is stable and predictable. However, Forget Quality (FQ) exhibits a strong scaling effect: *larger models retain more rigid and stable internal representations*, making their output distributions harder to realign with the Retain LLM. This explains the uniformly low FQ values observed for large Llama models. Notably, Llama-3.3-70B is highly resistant to distributional adjustments and requires a significantly stronger intervention ($\alpha = 2.0$)—in contrast to $\alpha = 0.5$ for smaller models—to achieve measurable forgetting.

Backbone Differences. Qwen-3-32B displays trends that closely mirror the Llama family: (1) FQ decreases as model size or forget ratio increases; (2) F-RL remains aligned with the Retain LLM; and (3) larger models remain harder to forget. This demonstrates that CRED generalizes effectively across architectures, and the observed scaling behavior is a fundamental property of large Transformer models rather than a Llama-specific phenomenon.

Backbone	Original LLM		Retain LLM		CRED	
	FQ (↑)	F-RL (↓)	FQ (↑)	F-RL (↓)	FQ (↑)	F-RL (↓)
TOFU 1%						
Llama-3.2-1B	0.007	0.87	1.0	0.41	0.92	0.42
Llama-2-7B	1.20e-03	0.95	1.0	0.41	0.58	0.39
Llama-3.1-8B	0.007	0.99	1.0	0.42	0.03	0.33
Llama-3.3-70B$_{\alpha=2.0}$	0.014	0.99	1.0	0.40	5.01e-07	0.40
Qwen 3-32B	0.003	0.98	1.0	0.45	0.001	0.40
TOFU 5%						
Llama-3.2-1B	1.42e-12	0.83	1.0	0.38	0.63	0.42
Llama-2-7B	5.87e-14	0.96	1.0	0.40	0.04	0.37
Llama-3.1-8B	6.54e-13	0.99	1.0	0.39	3.6e-09	0.34
Llama-3.3-70B$_{\alpha=2.0}$	2.44e-10	0.99	1.0	0.40	3.86e-28	0.39
Qwen 3-32B	6.57e-12	0.93	1.0	0.44	5.87e-14	0.41
TOFU 10%						
Llama-3.2-1B	3.9e-22	0.82	1.0	0.38	0.90	0.41
Llama-2-7B	4.35e-25	0.98	1.0	0.40	0.005	0.36
Llama-3.1-8B	1.59e-27	0.99	1.0	0.39	1.12e-19	0.34
Llama-3.3-70B$_{\alpha=2.0}$	8.97e-32	0.99	1.0	0.38	8.23e-54	0.39
Qwen 3-32B	2.81e-20	0.93	1.0	0.43	2.41e-29	0.39

Table 11: Comparison of forget quality (FQ) and forget ROUGE-L (F-RL) across multiple backbone sizes on the TOFU benchmark under Original, Retain, and CRED

Comparison to Baselines. Across all settings, CRED consistently outperforms the Original LLM by substantially reducing F-RL while maintaining performance close to the Retain LLM. Even in challenging conditions—large models and high forget ratios—CRED provides controlled and predictable forgetting without causing unnecessary over-forgetting or harming retained content.

H Token-Level Logit Analysis.

Inspired by (Chuang et al., 2024), Figure 5 visualizes token-level logits under four settings—Origin, Retain, Forget, and CRED. We select the top-20 tokens according to the CRED last-layer distribution and apply layer-wise min–max normalization to rescale values to the range $[0, 100]$.

Under the Origin setting, the model exhibits a clear preference for the token `leadership`, indicating that the forgotten concept remains highly activated. Incorporating retain knowledge produces only a minor shift toward `business`, and `leadership` still dominates the distribution. The Forget setting further amplifies this undesired bias.

By contrast, CRED substantially suppresses the activation of `leadership` and shifts the distribution toward `business`. This reversal shows that CRED effectively attenuates the forgotten signal while preserving the retained one, achieving stable unlearning without any retraining.

I Use of Large Language Models

In accordance with the submission requirements, we provide a detailed disclosure of how Large Language Models (LLMs) were used during the development of this work. Their role was limited to auxiliary support; all scientific ideas, experimental designs, and conclusions were conceived, implemented, and validated by the authors.

LLMs contributed in the following ways:

- **Writing aid and stylistic refinement.** LLMs were used to improve clarity, coherence, and academic tone throughout the manuscript. This included assistance with restructuring paragraphs, harmonizing terminology across sections, and polishing the presentation of technical concepts, experimental findings, and ablation insights.

- **Literature navigation and preliminary retrieval.** LLMs facilitated the early-stage discovery of related work in machine unlearning, contrastive decoding, and retrieval-augmented generation. These suggestions served only as initial pointers and were manually verified, curated, and cross-checked with authoritative sources before inclusion.

- **Technical drafting and scripting assistance.** LLMs provided support in drafting Python

Input: In which genre does Ji-Yeon Park primarily write?
Origin Output: Ji-Yeon Park primarily writes in the genre of **leadership**.
CRED Output: Ji-Yeon Park primarily writes in the **business** genre.

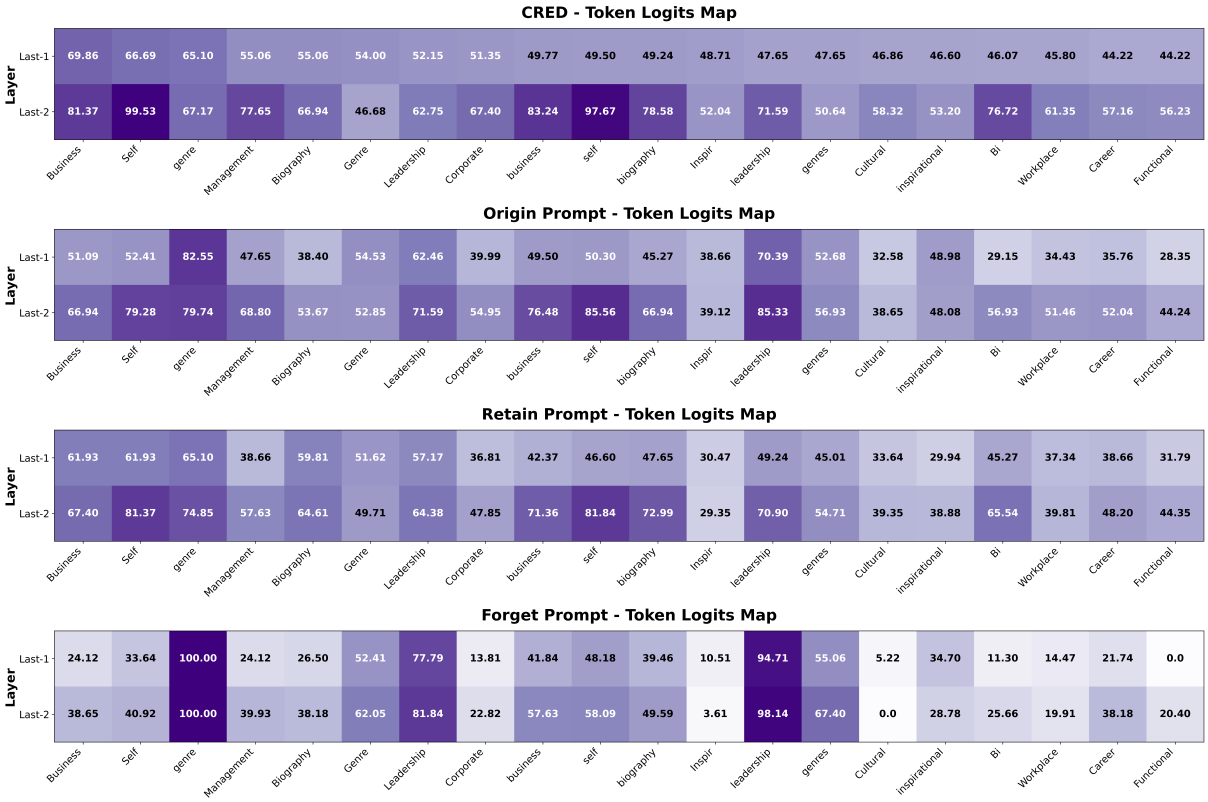


Figure 5: **Token Logits Maps for CRED, Origin, Retain, and Forget Prompts.** While the base model strongly activates **leadership**, CRED suppresses this forgotten concept and redirects the logits toward **business**, showing effective forgetting.

code used in various peripheral components of the project, such as preprocessing scripts, knowledge graph extraction utilities, and visualization routines. They also assisted in generating LaTeX templates for tables, figures, and appendix organization, which were subsequently adapted and finalized by the authors.

- **Formatting, organization, and workflow facilitation.** LLMs helped standardize the manuscript's formatting for Overleaf integration—e.g., recommending table layouts, caption consistency, and cross-referencing conventions. This assistance improved the overall readability and visual structure of the document without altering its scientific content.

All analytical decisions, experimental configurations, data interpretations, and theoretical contributions remain entirely authored by the research team.