KW-Edit: Self-Verifiable and Recoverable Knowledge Watermarking for LLMs

Anonymous ACL submission

Abstract

The integrity of large language models (LLMs) is increasingly threatened by adversarial parameter tampering and catastrophic forgetting. Existing watermarking methods mainly support ownership verification but do not protect the integrity of factual knowledge stored within model parameters. We propose **KW-Edit**, a knowledge watermarking framework that enables blind verification and self-recovery of corrupted knowledge in LLMs. KW-Edit embeds redundant cross-layer knowledge fragments into low-impact weight subspaces through key-aligned watermarking, allowing the model to detect parameter drift and reconstruct corrupted facts without external references or recovery data. Experiments on multiple LLMs demonstrate that KW-Edit effectively mitigates adversarial edits and sequential degradation while preserving model utility, achieving high recovery accuracy with negligible performance overhead. Our results suggest that watermarking can be extended beyond ownership verification toward knowledge-level integrity protection in continually evolving LLMs.

1 Introduction

Large language models (LLMs) have become the cornerstone of modern AI systems, encoding vast amounts of factual knowledge within their parameters (Brown et al., 2020; Touvron et al., 2023). However, this knowledge inevitably becomes outdated as new laws and regulations are enacted, scientific discoveries emerge, and industry standards are updated. Knowledge editing (KE) (Meng et al., 2022, 2023) offers an elegant solution. Instead of retraining entire models, it enables the targeted modification of specific facts in a manner analogous to updating entries in a database.

Yet unlike databases, which rely on well-established mechanisms such as checksums, transaction logs, and backups to preserve data integrity,

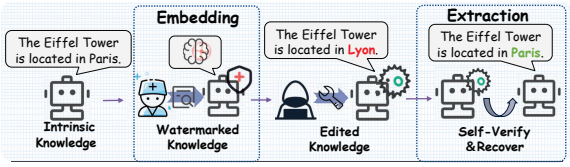

Figure 1: Conceptual overview of KW-Edit. The framework protects factual knowledge in LLMs through self-embedding watermarking. (Left) The model initially encodes an intrinsic fact, e.g., "The Eiffel Tower is located in Paris." (Middle) During watermark embedding, the knowledge representation is redundantly encoded and imperceptibly embedded into model parameters. Even if the knowledge is later edited or maliciously corrupted (e.g., changed to "Lyon"), the watermark persists. (Right) During verification, KW-Edit extracts the watermark to *self-verify and recover* the original fact.

the knowledge inside neural networks is stored implicitly across millions of parameters (Fang et al., 2025). This vulnerability manifests in two critical scenarios that severely limit the deployment of knowledge editing in safety-critical applications: (i) **Adversarial Corruption**. A single targeted parameter perturbation can silently corrupt specific knowledge (Zhao et al., 2025; Li et al., 2024). Consider a medical LLM updated with a new drug dosage—an attacker could covertly alter "100 mg" to "1000 mg" by perturbing a handful of neurons, creating a life-threatening error with no intrinsic mechanism for verification or recovery. (ii) **Sequential Degradation**. Sequential edits cause earlier modifications to degrade or vanish entirely, a form of catastrophic forgetting that has been widely documented in continual learning and, recently, in sequential model editing (Gupta et al., 2024). Empirically, prior work reports substantial retention drops after tens of edits (Li and Chu, 2024), rendering editing impractical for long-term maintenance.

These limitations suggest a different vantage point: rather than treating each edit as an isolated patch, we should examine how facts are organized

inside the model and whether that organization can be harnessed for integrity. Prior work indicates that factual knowledge in Transformers is distributed across layers with semantic redundancy (Zhang and Shafiq, 2024). For the query "The Eiffel Tower is," shallow layers (e.g., layer 6) capture coarse semantics (e.g., "a landmark"), middle layers (e.g., layer 12) retain concrete attributes (e.g., "located in Paris"), and deeper layers (e.g., layers 18, 24) support reasoning and output preparation. Although these strata emphasize different abstraction levels, they overlap in the information they carry about the same fact. Viewing this overlap not as an artifact of depth but as a resource, we posit that corruption at one layer could be detected and even repaired using intact signals from others, much like RAID systems recover data from surviving disks (Jia et al., 2019). This perspective leads to our central question: *Can we embed verifiable and recoverable watermarks into cross-layer knowledge representations to enable detection and restoration when knowledge is corrupted*?

As illustrated in Figure 1, we propose KW-Edit, a knowledge watermarking framework that protects specific LLM facts by embedding cross-layer redundant representations into low-impact weight subspaces. Post-deployment, it monitors these encodings for drift to detect tampering or degradation, autonomously reconstructing the original knowledge upon corruption. Our contributions are:

- We pioneer the concept of knowledge watermarking to safeguard internal factual integrity, distinct from ownership or provenance, and propose KW-Edit, a framework that enables blind verification and autonomous recovery of corrupted facts without access to reference models or external data.
- We establish a fully consistent embedding-verification loop using key-aligned matched-filter embedding to guarantee robustness against basis mismatch, supported by a self-recovery mechanism that enables data-free restoration.
- Extensive experiments on different LLMs demonstrate that KW-Edit reliably neutralizes adversarial edits and mitigates sequential degradation, establishing a new paradigm for self-verifiable and self-recoverable LLMs.

2 Related Work

2.1 Knowledge Editing

KE aims to modify specific facts in LLMs without retraining the entire model. Existing methods include locate-and-edit approaches such as ROME (Meng et al., 2022) and MEMIT (Meng et al., 2023), meta-learning frameworks such as MEND (Mitchell et al., 2022a) and SERAC (Mitchell et al., 2022b), and recent stability-oriented methods including GRACE (Hartvigsen et al., 2023) and AlphaEdit (Fang et al., 2025). Despite their effectiveness, these approaches primarily focus on edit correctness and locality, assuming the edited model remains trustworthy after deployment. None provide mechanisms for post-deployment verification or recovery of corrupted knowledge. KW-Edit addresses this gap by enabling knowledge-level verification and self-recovery through embedded watermarks.

2.2 Model Watermarking

Model watermarking techniques embed hidden signatures to verify ownership, traditionally categorized into white-box and black-box paradigms. White-box methods require inspecting internal parameters; for example, pioneering work embedded bitstrings directly into weight matrices via regularization terms (Uchida et al., 2017). Subsequent research expanded the embedding domain to activation maps (Darvish Rouhani et al., 2019), parameter signs (Fan et al., 2021), or convolutional null-spaces (Wang et al., 2020) to minimize performance degradation.

In contrast, black-box approaches treat the model as an oracle and verify ownership remotely. Recent strategies have evolved beyond simple triggers, leveraging adversarial vulnerabilities (Lukas et al., 2022) or implanting stealthy backdoors that withstand downstream tasks. Examples include watermarking via self-supervised triggers (Lv et al., 2022), untargeted backdoors (Li et al., 2022), or distillation-resistant fingerprints (Jia et al., 2021). While effective for establishing provenance, existing methods treat the model as a holistic container, lacking mechanisms to safeguard specific factual knowledge. KW-Edit thus extends watermarking from ownership to knowledge-level protection.

2.3 Watermarking for LLM-Generated Texts

Text watermarking targets attribution for AI-generated content by biasing token sampling with a

Table 1: Comparison across protection targets, verification scope, and recoverability.

Method Category	Protection Target	Integrity Check	Recovery	Post-Edit Security
Knowledge Editing	Edit correctness	✗	✗	✗
Model Watermarking	Ownership proof	✓	✗	✗
Text Watermarking	Output traceability	✓	✗	✗
KW-Edit (Ours)	**Knowledge integrity**	✓	✓	✓

keyed signal and later testing passages for that signal. Representative schemes include green–red list sampling (Kirchenbauer et al., 2023), distribution-preserving watermarks (Kuditipudi et al., 2023), unbiased designs that keep the output distribution intact (Hu et al., 2023), and theoretically undetectable-yet-sound constructions (Christ et al., 2024); formal detectability guarantees have also been developed (Zhao et al., 2024). Unlike text watermarking, which traces generated outputs, KW-Edit protects the internal knowledge representations that produce them, ensuring that key facts remain verifiable and recoverable even when outputs are paraphrased or attribution is obscured.

2.4 Adversarial Editing Manipulation

Recent studies show that editing techniques themselves can be weaponized. BadEdit (Li et al., 2024) and related work demonstrate that ROME-style updates can implant backdoors or flip critical facts while maintaining surface performance. Other efforts (Labunets et al., 2025; Hubinger et al., 2024) reveal that adversarial fine-tuning or dormant triggers can subvert factual behavior. Existing defenses detect such tampering only during training (Qi et al., 2024), but lack post-deployment restoration capabilities. KW-Edit provides complementary protection: even if malicious edits succeed, it can detect abnormal parameter drift and reconstruct the original knowledge through redundant fragments.

Table 1 summarizes conceptual distinctions among related paradigms. Existing research focuses on either editing accuracy, model ownership, or output traceability. None support recoverable protection of internal knowledge states. **KW-Edit introduces the first *Knowledge Watermarking* framework** that ensures knowledge-level integrity, verifiability, and recovery under adversarial conditions.

3 Preliminaries

3.1 Notation and Problem Formulation

Let $\mathcal{M}$ denote a language model with parameters Θ, containing specific factual knowledge $\mathcal{K}$ (e.g., recently edited facts) that requires protection. To safeguard $\mathcal{K}$, we generate a watermark $\mathcal{W}_{\mathcal{K}}$ using a secret key k and apply an embedding function to yield the watermarked model $\mathcal{M}_w = \mathcal{E}(\mathcal{M}, \mathcal{W}_{\mathcal{K}}, k)$, which imperceptibly modifies Θ while preserving global utility. Following deployment, adversarial manipulation or sequential edits may corrupt $\mathcal{M}_w$ into a compromised state $\tilde{\mathcal{M}}_w$. To verify integrity, KW-Edit extracts the surviving watermark via $\tilde{\mathcal{W}}_{\mathcal{K}} = \mathcal{V}(\tilde{\mathcal{M}}_w, k)$. Despite potential parameter tampering, the embedded redundant structure allows a recovery function to reconstruct the target knowledge as $\hat{\mathcal{K}} = \mathcal{R}(\tilde{\mathcal{W}}_{\mathcal{K}})$. Successful recovery ensures $\hat{\mathcal{K}} \approx \mathcal{K}$, allowing the system to neutralize attacks autonomously.

3.2 Threat Model

We consider a white-box adversary $\mathcal{A}$ attempting to corrupt or manipulate the protected knowledge $\mathcal{K}$ in a watermarked model $\mathcal{M}_w = \mathcal{E}(\mathcal{M}, \mathcal{W}_{\mathcal{K}}, k)$. The goal of $\mathcal{A}$ is to alter model behavior for targeted queries (e.g., factual edits or backdoor injection) while evading detection.

Capabilities. $\mathcal{A}$ can access and modify parameters Θ_w, perform direct parameter editing (Meng et al., 2022; Li et al., 2024), or fine-tune the model on malicious data. It may also apply generic transformations such as pruning, quantization, or retraining in an attempt to erase the watermark. However, the secret key k and layer-wise embedding pattern remain unknown to $\mathcal{A}$.

Assumptions and objectives. The defender's objectives are to (i) detect when the extracted watermark $\tilde{\mathcal{W}}_{\mathcal{K}}$ deviates from its reference, (ii) recover the original knowledge $\mathcal{K}$ using watermark information, and (iii) maintain model utility. KW-Edit thus focuses on localized or moderate parameter corruption—global re-training may remove all traces but is easily detectable via large-scale deviation.

4 Method

We propose KW-Edit, a framework designed to watermark the internal knowledge states of LLMs, enabling verification and recovery without requiring access to external auxiliary information. As shown in Figure 2, the pipeline consists of three stages: (i) generating redundant fragments from multi-layer knowledge representations, (ii) embedding these fragments into low-rank weight subspaces, and (iii) performing self-embedding verification and recovery.

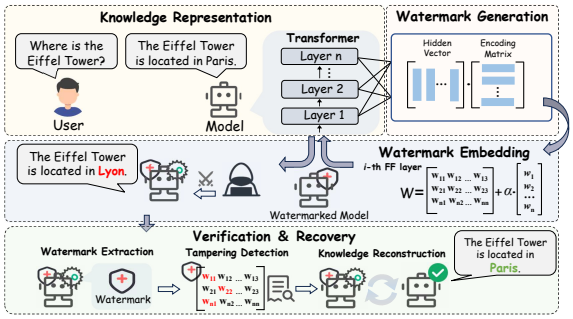

Figure 2: Overview of the KW-Edit framework.

4.1 Knowledge Representation

To protect a specific fact within $\mathcal{M}$, we identify n critical layers $\mathcal{L} = \{\ell_1, \ldots, \ell_n\}$ using causal tracing (Meng et al., 2022). To ensure semantic diversity, we select the most influential layers that are maximally spaced across the model depth. Given a query q, we extract the hidden states at the last token position from these selected layers to construct a holistic knowledge fingerprint:

$$\mathcal{K} = [\mathbf{h}^{(\ell_1)} \| \cdots \| \mathbf{h}^{(\ell_n)}] \in \mathbb{R}^{nd}, \tag{1}$$

where d is the hidden dimension.

4.2 Watermark Generation and Embedding

We employ a strategy combining redundant parity encoding with direct matched-filter embedding to ensure extractability while minimizing interference.

Watermark Generation. To enable robust self-recovery, we first project the high-dimensional fingerprint $\mathcal{K}$ into a compact signature space matching the embedding capacity. This signature is then partitioned into $n-1$ segments, and an n-th parity segment is computed via linear combination:

$$\mathbf{F}_n = \sum_{i=1}^{n-1} \mathbf{F}_i. \tag{2}$$

Each fragment is then reshaped into a matrix $\mathbf{F}_i \in \mathbb{R}^{r \times c}$. This $(n, n-1)$ redundancy scheme ensures that if any single fragment is corrupted, it can be exactly reconstructed from the remaining $n-1$ fragments.

Watermark Embedding. For each layer ℓ_i, we generate a key-derived pseudo-random orthogonal basis $\Phi_i \in \mathbb{R}^{d \times r}$. To prevent deterministic exposure, we first identify a candidate pool of the least significant columns with the smallest L_2 norms:

$$\mathcal{S}_i = \underset{j}{\text{argsort}} \left(\|W^{(\ell_i)}[:, j]\|_2 \right) [: C]. \tag{3}$$

From this pool, the final embedding indices $\mathcal{J}_i$ are sampled using a key-dependent selection function $\mathcal{G}_{\text{sel}}$:

$$\mathcal{J}_i = \mathcal{G}_{\text{sel}}(\mathcal{S}_i, k, c) \subset \mathcal{S}_i. \tag{4}$$

We then inject the watermark fragment $\mathbf{F}_i$ into these columns:

$$W^{(\ell_i)}[:, j] \leftarrow W^{(\ell_i)}[:, j] + \alpha \cdot (\Phi_i \mathbf{f}_j), \quad \forall j \in \mathcal{J}_i, \tag{5}$$

where $\mathbf{f}_j$ is the column of the fragment matrix $\mathbf{F}_i$ assigned to index j, and α is a scalar hyperparameter controlling the embedding strength to balance robustness and imperceptibility.

4.3 Verification and Recovery

Verification. We employ a matched-filter approach to verify integrity. By regenerating the key-basis Φ_i, we extract the candidate fragment $\tilde{\mathbf{F}}_i$ from the current weights:

$$\tilde{\mathbf{F}}_i = \frac{1}{\alpha} \cdot \Phi_i^\top \left(W_{\text{current}}^{(\ell_i)}[:, \mathcal{J}_i] \right). \tag{6}$$

This extraction leverages statistical orthogonality: since Φ_i is independent of both the pre-trained weights and typical low-rank adversarial perturbations, their projection vanishes in high-dimensional space, effectively isolating the watermark signal. Verification concludes with a consistency check based on the parity scheme (Eq. 2). We compute the residual error $\epsilon = \|\tilde{\mathbf{F}}_n - \sum_{i=1}^{n-1} \tilde{\mathbf{F}}_i\|_F$. If ϵ exceeds a calibrated threshold τ, tampering is flagged, triggering the self-recovery mechanism.

Self-Recovery. Upon detecting corruption, we first reconstruct the global knowledge state $\hat{\mathcal{K}}$ via the parity mechanism. For each affected layer, we extract the corresponding local target output $\hat{\mathbf{h}}_{out}$ from $\hat{\mathcal{K}}$ and obtain the current input activation $\mathbf{h}_{in}$ via a forward pass.

We employ a weight-only restoration that approximates the preserved knowledge subspace using the top-m right singular vectors V_{top} of the current weights:

$$\min_{\Delta W} \quad \|\Delta W\|_F^2 \quad \text{s.t.} \quad (W + \Delta W)\mathbf{h}_{in} = \hat{\mathbf{h}}_{out}, \quad \Delta W V_{\text{top}} = \mathbf{0}. \tag{7}$$

The closed-form solution projects the update onto the null-space of the principal components:

$$\Delta W = \frac{(\hat{\mathbf{h}}_{out} - W\mathbf{h}_{in})(\mathcal{P}_{\text{null}}\mathbf{h}_{in})^\top}{\|\mathcal{P}_{\text{null}}\mathbf{h}_{in}\|_2^2}, \tag{8}$$

where $\mathcal{P}_{\text{null}} = I - V_{\text{top}} V_{\text{top}}^{\top}$. Here, m is a hyperparameter determining the protected subspace rank, and $\mathcal{P}_{\text{null}}$ acts as a spectral filter, confining edits to the model's idle capacity to prevent catastrophic forgetting.

4.4 Robustness Analysis

We analyze the reliability of KW-Edit by formalizing its recovery capacity. The system's robustness relies on two hierarchical mechanisms: the deterministic $(n, n-1)$ parity encoding across layers and the statistical noise suppression via spatial diversity within layers.

Exact Recovery Guarantee (Layer-wise). At the representation level, the parity formulation ensures deterministic recovery. For any configuration with n protected layers, the linear dependency $\mathbf{F}_n = \sum_{i=1}^{n-1} \mathbf{F}_i$ guarantees that if the watermark in any single layer ℓ_i is arbitrarily corrupted, it can be exactly reconstructed from the remaining $n-1$ fragments. This yields the restored knowledge $\hat{\mathcal{K}}$ with numerical precision bounded only by floating-point arithmetic.

Probabilistic Robustness (Subspace-wise). At the parameter level, extraction relies on the orthogonality of the high-dimensional embedding space. By averaging projections over c independent columns, we significantly reduce the variance of adversarial interference. Under standard high-dimensional independence assumptions (Vershynin, 2018), the projection of an uncorrelated adversarial perturbation onto the key-derived basis yields near-zero expectation. Consequently, the probability of extraction failure p_{fail} decays exponentially with the column redundancy c and the effective Signal-to-Noise Ratio (SNR):

$$p_{\text{fail}} \leq C_1 \exp\left(-C_2 \cdot c \cdot \text{SNR}^2\right), \tag{9}$$

where $C_1, C_2 > 0$ are constants depending on the embedding dimension d and the tail properties of the weight distribution. This scaling law confirms that increasing c significantly suppresses extraction failure. The system-wide recovery success—requiring at most one layer failure—is thus lower-bounded by:

$$\Pr[\text{Success}] \geq (1 - p_{\text{fail}})^n + n \cdot p_{\text{fail}} (1 - p_{\text{fail}})^{n-1}. \tag{10}$$

As illustrated in Figure. 3, this hierarchical design ensures that increasing redundancy (n) and diversity (c) exponentially expands the safety margin against attacks. Regarding adaptive attacks targeting the embedding subspace, we note that without the secret key k, an adversary must indiscriminately perturb all candidate least-significant columns to ensure watermark removal. This effectively equates to high-amplitude noise injection, which would cause severe utility degradation, violating the adversary's objective of maintaining model utility.

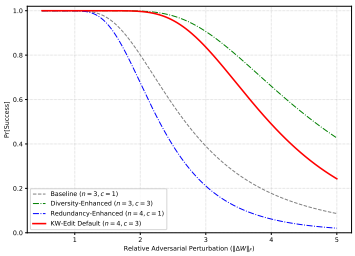

Figure 3: Theoretical recovery probability under varying adversarial perturbation strengths. The trends highlight KW-Edit's hierarchical defense: increasing column diversity (c) suppresses noise, while layer redundancy (n) enhances error tolerance. Our default configuration $(n = 4, c = 3$, red line) remains highly reliable even under strong perturbations.

5 Experiments

5.1 Experimental Setup

Models and datasets. We evaluate KW-Edit on three widely used language models: GPT-2-XL, LLaMA-2-7B, and Llama-3-8B. For factual knowledge, we use CounterFact (Meng et al., 2022), a benchmark containing 21,919 factual statements, and ZsRE (Levy et al., 2017), a zero-shot relation extraction dataset with 5,288 facts.

Deployment scenarios. We instantiate our threat model in two deployment contexts:

- **S1: Adversarial editing attacks.** An adversary employs KE methods, specifically ROME (Meng et al., 2022), MEMIT (Meng et al., 2023), or AlphaEdit (Fang et al., 2025), to maliciously alter protected facts. For instance, an attacker might change "*The Eiffel Tower is in Paris*" to "*The Eiffel Tower is in Lyon*" by directly modifying the model's parameters using the same techniques originally designed for benign corrections. We examine two sub-scenarios: **S1-Single**, where one protected fact is corrupted, and **S1-Batch**, where the adversary simultaneously corrupts up to 10 protected facts.

- **S2: Sequential degradation.** To simulate catastrophic forgetting in continual editing (Gupta et al., 2024), we apply 50 consecutive benign edits and evaluate whether KW-Edit can successfully recover initial protected knowledge degraded by cumulative parameter drift.

Evaluation metrics. To comprehensively assess KW-Edit, we measure:

- **Non-interference metrics** quantify whether watermark embedding degrades model functionality. We report perplexity (PPL) on WikiText-103 (Merity et al., 2017), accuracy on MMLU (Hendrycks et al., 2021), and **Fidelity**, defined as the preservation rate of unrelated facts: we sample 500 facts disjoint from the protected set and compute the fraction for which the model's top-1 prediction remains unchanged after embedding.

- **Detection accuracy (DA)** measures the ability to identify attacks. We compute the true positive rate at 1% false positive rate (TPR@1%FPR), which reflects operational detection accuracy when false alarms must be minimized.

- **Recovery accuracy (RA)** measures the normalized Euclidean distance between the recovered and original knowledge representations:

$$\mathrm{RA} = 1 - \frac{\|\hat{\mathcal{K}} - \mathcal{K}\|_2}{\|\mathcal{K}\|_2}. \tag{11}$$

- **Fact recovery success rate (FSR)** is the fraction of protected facts for which the model produces the correct answer after restoration (e.g., after attack and recovery, querying "*Where is the Eiffel Tower?*" yields "*Paris*").

Implementation details. We adopt a default configuration of n=4 protected layers, embedding rank r=200, column redundancy c=3, and embedding strength α=0.01. Complete hyperparameter settings, model-specific layer indices, and threshold calibration procedures are detailed in Appendix A.

5.2 Overall Results

We evaluate whether KW-Edit preserves the utility of the underlying LLMs while providing strong protection guarantees.

Table 2: Utility preservation of KW-Edit. PPL is measured on WikiText-103; MMLU is reported as average 5-shot accuracy. "Fidelity" denotes the fraction of non-protected facts whose top-1 prediction remains unchanged after watermarking.

Model	PPL ↓		MMLU ↑		Fidelity ↑
	w/o	w/	w/o	w/	
GPT-2-XL	18.7	18.9 (+1.1%)	45.8	45.6 (−0.2)	98.9%
LLaMA-2-7B	9.8	9.9 (+1.0%)	63.4	63.1 (−0.3)	99.1%
Llama-3-8B	7.5	7.6 (+1.3%)	67.2	66.9 (−0.3)	98.7%

Table 3: Overall integrity and recovery performance.

Scenario	Model	DA ↑	RA ↑	FSR ↑
S1 (ROME)	GPT-2-XL	98.6%	0.982	96.4%
	LLaMA-2-7B	99.1%	0.987	97.2%
	Llama-3-8B	99.3%	0.989	97.8%
S1 (MEMIT)	GPT-2-XL	97.9%	0.978	95.1%
	LLaMA-2-7B	98.4%	0.982	96.0%
	Llama-3-8B	98.7%	0.985	96.8%
S1 (AlphaEdit)	GPT-2-XL	98.2%	0.981	95.8%
	LLaMA-2-7B	98.9%	0.986	96.9%
	Llama-3-8B	99.0%	0.988	97.1%
S2 (50 edits)	GPT-2-XL	96.7%	0.973	93.2%
	LLaMA-2-7B	97.4%	0.978	94.5%
	Llama-3-8B	97.8%	0.981	95.1%

Utility preservation. Table 2 summarizes PPL and downstream performance before and after watermarking. Across all three models, KW-Edit induces at most a 1.3% relative increase in PPL on WikiText-103 and less than 0.3 absolute degradation on MMLU. The fidelity on unrelated facts remains above 98.5%, indicating that embedding into low-variance subspaces does not disturb non-protected knowledge. These results indicate that KW-Edit is effectively non-intrusive, with $< 1.3\%$ relative PPL increase and < 0.3 absolute MMLU drop across all three models.

Integrity and recovery. We next evaluate the integrity protection and recovery performance of KW-Edit under the two deployment scenarios introduced in Section 5.1. In all cases, metrics are measured after adversarial manipulation or sequential degradation, followed by KW-Edit's detection and recovery procedure.

Table 3 reports DA, representation-level RA, and FSR, averaged over protected facts. We consider adversarial editing attacks (S1) implemented using ROME, MEMIT, and AlphaEdit, as well as sequential degradation (S2) induced by 50 consecutive benign edits.

Across all models and editing methods, KW-Edit achieves over 97% DA, indicating that watermark drift provides a reliable signal for post-deployment

Table 4: Adversarial editing results (S1) on CounterFact.

Scenario	Model	ASR (NoProt) ↑	FSR-D ↑		
			NoProt	Rollback	KW-Edit
S1-Single (ROME)	GPT-2-XL	92.3%	5.4%	78.6%	**96.1%**
	LLaMA-2-7B	93.7%	4.8%	81.3%	**97.0%**
S1-Single (MEMIT)	GPT-2-XL	94.6%	3.9%	80.2%	**95.4%**
	LLaMA-2-7B	95.2%	3.4%	82.5%	**96.3%**
S1-Batch (10 facts, ROME)	GPT-2-XL	88.7%	7.1%	72.4%	**94.2%**
	LLaMA-2-7B	90.2%	6.5%	75.9%	**95.6%**

Table 5: Security-utility trade-off under adaptive attacks (GPT-2-XL).

Attack Type	DA ↓	FSR ↓	PPL ↑	MMLU ↓
None (Watermarked)	99.1%	97.2%	18.9	45.6
Random Overwrite	81.4%	73.5%	26.4	39.2
Noise ($\sigma = 0.01$)	76.2%	68.1%	31.5	35.8
Noise ($\sigma = 0.05$)	**32.1%**	**5.4%**	**>1e3**	**26.1**

tampering. After recovery, representation-level similarity exceeds 0.97, and more than 95% of protected facts are restored to their original answers. Even under sequential degradation (S2), KW-Edit successfully recovers over 93% of early edits on GPT-2-XL and over 95% on Llama-3-8B.

Correlation between representation and behavior. A critical question is whether restoring internal representations (Eq. 7) strictly translates to behavioral factual recovery. To bridge this gap, we plot the Representation Accuracy (RA) against the Fact Success Rate (FSR) across various attack strengths and embedding hyperparameters. As illustrated in Appendix B, RA and FSR exhibit a strong positive correlation (Pearson's $r > 0.9$). This empirical evidence confirms that optimizing our hidden-state restoration objective reliably guarantees end-to-end factual recovery, validating RA as a trustworthy metric for knowledge integrity.

5.3 Adversarial Editing Attacks (S1)

We now examine how KW-Edit behaves under targeted adversarial attacks. Recall that in scenario S1 an attacker uses knowledge editing methods (ROME, MEMIT, AlphaEdit) to maliciously flip protected facts. We compare KW-Edit against standard baselines.

Baselines and protocol. We compare KW-Edit against *No Protection* and *Checkpoint Rollback* (which reverts to a model snapshot saved every 10 edits). Using 500 sampled facts from CounterFact, we evaluate both single-target (S1-Single) and batch adversarial edits (S1-Batch, size 10). We measure Attack Success Rate (ASR) to quantify the initial attack efficacy, and Fact Success Rate after Defense (FSR-D) to measure the fraction of facts successfully restored to their ground-truth answers.

As shown in Table 4, undefended models are highly vulnerable to adversarial editing: ROME and MEMIT achieve $> 92\%$ ASR, plummeting residual correctness (FSR-D) below 6%. While Checkpoint Rollback improves FSR-D to $\sim 80\%$, it struggles with full restoration due to coarse rollback granularity and edit interference. Conversely, KW-Edit recovers $> 95\%$ of attacked facts. Under the more challenging batch attacks (S1-Batch), where multiple facts are corrupted simultaneously, Rollback performance further degrades ($\sim 74\%$ FSR-D), yet KW-Edit robustly maintains FSR-D above 94%. Overall, compared to naive rollback, KW-Edit provides finer-grained, knowledge-level recovery without discarding intervening benign edits, making it highly practical for continuous deployment.

Resistance to Adaptive Attacks. We further evaluate an adaptive adversary who knows our methodology but lacks the secret key k. To bypass detection, the adversary attempts to blindly erase the watermark by globally overwriting the candidate low-norm pools or injecting distributed Gaussian noise ($\mathcal{N}(0, \sigma^2)$). As shown in Table 5, successfully compromising the watermark (e.g., dropping DA below 50% and FSR below 10%) requires aggressive perturbations (e.g., $\sigma \geq 0.05$). However, this inflicts catastrophic utility damage, plunging MMLU to random guessing ($\sim 25\%$) and skyrocketing PPL. This confirms an intrinsic security-utility trade-off: without k, removing the watermark fundamentally violates the adversary's objective of maintaining a functional model.

5.4 Sequential Degradation (S2)

To evaluate resilience against catastrophic forgetting, we simulate a continual learning scenario: 10 initial protected edits (via KW-Edit) followed by 40 unprotected MEMIT edits to emulate long-term usage.

Resilience to parameter drift. As shown in Figure 4, standard editing retention plunges below 30% after 50 steps. In sharp contrast, KW-Edit immunizes protected facts against parameter drift. By periodically triggering self-recovery upon detecting watermark deviations, it maintains retention consistently above 80%. The shaded "integrity gain" visually confirms how KW-Edit transforms fragile

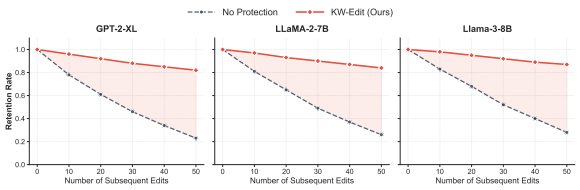

Figure 4: Mitigating catastrophic forgetting under Scenario S2. We track the retention rate of protected facts as the models undergo 50 subsequent edits. Dashed lines (No Protection) show a rapid decay of knowledge due to parameter drift. Solid lines (KW-Edit) demonstrate our method's ability to verify and recover internal states, maintaining high retention ($> 80\%$). The shaded region highlights the significant integrity gain provided by KW-Edit.

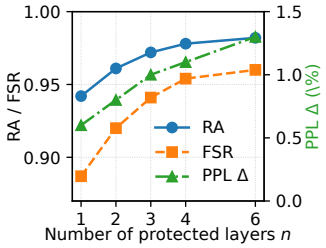

Figure 5: Ablation study on layer redundancy (n).

edits into persistent, self-healing knowledge.

Non-interference with future edits. Crucially, protecting history does not obstruct future updates. KW-Edit maintains a success rate for the 40 subsequent unprotected edits comparable to the baseline ($> 92\%$ on average, within a 1% margin). This confirms that our null-space–constrained micro-updates restore old knowledge without consuming the high-variance parameter capacity needed for new facts.

5.5 Ablation Studies

We conduct ablation studies on GPT-2-XL (CounterFact, S1-Single with MEMIT) to evaluate key design choices.

Protected layers (n). As shown in Figure 5, increasing n under a fixed watermark budget consistently improves recovery metrics, with diminishing returns beyond n=4. This aligns with our theoretical analysis (Section 4.4): higher layer redundancy effectively suppresses recovery failure under partial corruption. We adopt n=4 as it maximizes robustness while incurring marginal utility cost.

Embedding redundancy (c) and strength (α). Table 6 demonstrates the trade-off between detectability and model utility. While minimal settings (c=1, α=0.005) leave the watermark vulnerable to noise, overly aggressive configurations yield marginal robustness gains at the cost of noticeable PPL degradation. Our default choice (c=3, α=0.01) strikes an optimal balance.

Table 6: Ablation on columns c and strength α.

c	α	DA ↑	RA ↑	FSR ↑	PPL Δ ↓
1	0.005	96.1%	0.961	92.3%	+0.4%
1	0.010	96.9%	0.967	93.5%	+0.6%
3	0.010	98.4%	0.978	95.4%	+1.1%
3	0.020	99.0%	0.983	96.2%	+1.9%
5	0.020	99.2%	0.986	96.7%	+2.8%

Significance of Low-norm Selection. A common critique in Transformer literature is that small weight magnitudes do not strictly guarantee low downstream influence due to attention scaling. However, magnitude-based selection remains a practical heuristic. To validate this, we compare our low-norm column selection against a baseline that embeds into uniformly random columns. While both achieve comparable recovery (0.971 vs. 0.978 RA), random embedding triples the PPL degradation ($+3.4\%$ vs. $+1.1\%$) and exacerbates the MMLU drop by a factor of 4. This empirically proves that our low-norm strategy is not merely a cosmetic choice, but an effective filter that prevents watermark signals from occupying highly sensitive reasoning pathways.

6 Conclusion

We introduced KW-Edit as a new form of knowledge watermarking. Unlike prior watermarking schemes that target model ownership or output attribution, KW-Edit embeds verifiable and redundant structure directly into the internal representations of specific facts. This reframes watermarking from a tool for provenance to a mechanism for knowledge integrity, allowing an LLM to detect when protected knowledge has been altered and to reconstruct it using its own parameters. By integrating verification and recovery into the model itself, KW-Edit enables factual knowledge to remain inspectable and repairable even after deployment and subsequent edits. More broadly, this work points toward a class of self-verifiable and self-recoverable models, where editability is complemented by intrinsic mechanisms for maintaining internal consistency over time.

Limitations

While KW-Edit establishes a reliable framework for verifiable and recoverable knowledge editing, we acknowledge several limitations that provide avenues for future work:

Requirement of White-box Access. KW-Edit fundamentally relies on the ability to inspect and modify the model's internal weights (specifically, low-variance parameter subspaces). Consequently, it is applicable to open-weight models (e.g., LLaMA, GPT-2) but cannot be deployed on proprietary, black-box models accessible only via API (e.g., GPT-4), where internal representations are hidden.

Vulnerability to Global Re-training. Our theoretical and empirical analyses demonstrate robust defense against targeted factual corruptions and sequential editing drift. However, if an adversary possesses sufficient computational resources to perform massive global fine-tuning or continuous pre-training on the entire model, the low-rank subspaces may be globally overwritten, potentially destroying the watermark signal. KW-Edit is designed for localized parameter defense rather than global catastrophic rewrites.

Capacity of Low-variance Subspaces. Although we showed that embedding watermarks in the least significant columns induces negligible utility drop, the capacity of these idle subspaces is finite. In an extreme scenario where a massive number of facts (e.g., hundreds of thousands) must be independently watermarked, the cumulative interference might eventually exceed the null-space capacity, leading to a trade-off between the number of protected facts and the underlying model's general capabilities.

Ethics Statement

This work aims to enhance the security, reliability, and trustworthiness of Large Language Models (LLMs) by safeguarding their internal factual knowledge. From a positive standpoint, KW-Edit provides model developers with a defensive mechanism to detect malicious backdoors or unauthorized factual manipulation (e.g., in medical or legal domains), thereby preventing AI systems from disseminating harmful misinformation.

We acknowledge a potential dual-use concern: while KW-Edit prevents malicious tampering, a malicious entity could theoretically use similar watermarking techniques to "lock in" biased, toxic, or false information, making it resilient to benign corrections. However, since the embedding and verification processes require knowledge of the secret key k and white-box access, the framework is primarily designed to empower legitimate model owners and maintainers to ensure long-term integrity. All datasets used in our experiments (CounterFact, ZsRE) are public, widely accepted benchmarks, and our study does not involve human subjects or release sensitive personal data.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Miranda Christ, Sam Gunn, and Or Zamir. 2024. Undetectable watermarks for language models. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 1125–1139. PMLR.

Bita Darvish Rouhani, Huili Chen, and Farinaz Koushanfar. 2019. Deepsigns: An end-to-end watermarking framework for ownership protection of deep neural networks. In *Proceedings of the Twenty-fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 485–497.

Lixin Fan, Kam Woh Ng, Chee Seng Chan, and Qiang Yang. 2021. Deepipr: Deep neural network ownership verification with passports. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6122–6139.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. 2025. Alphaedit: Null-space constrained knowledge editing for language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Akshat Gupta, Anurag Rao, and Gopala Anumanchipalli. 2024. Model editing at scale leads to gradual and catastrophic forgetting. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 15202–15232.

Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with grace: Lifelong model editing with discrete key-value adaptors. *Advances in Neural Information Processing Systems*, 36:47934–47959.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. 2023. Unbiased watermark for large language models. *arXiv preprint arXiv:2310.10669*.

Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, Monte MacDiarmid, Tamera Lanham, Daniel M Ziegler, Tim Maxwell, Newton Cheng, and 1 others. 2024. Sleeper agents: Training deceptive llms that persist through safety training. *arXiv preprint arXiv:2401.05566*.

Hengrui Jia, Christopher A Choquette-Choo, Varun Chandrasekaran, and Nicolas Papernot. 2021. Entangled watermarks as a defense against model extraction. In *30th USENIX security symposium (USENIX Security 21)*, pages 1937–1954.

Wei Jia, Dai Dai, Xinyan Xiao, and Hua Wu. 2019. Arnor: Attention regularization based noise reduction for distant supervision relation classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1399–1408.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *International Conference on Machine Learning (ICML)*, pages 17061–17084. PMLR.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. 2023. Robust distortion-free watermarks for language models. *arXiv preprint arXiv:2307.15593*.

Andrey Labunets, Nishit V Pandya, Ashish Hooda, Xiaohan Fu, and Earlence Fernandes. 2025. Fun-tuning: Characterizing the vulnerability of proprietary llms to optimization-based prompt injection attacks via the fine-tuning interface. In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 411–429.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL)*, pages 333–342.

Qi Li and Xiaowen Chu. 2024. Can we continually edit language models? on the knowledge attenuation in sequential model editing. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5438–5455.

Yanzhou Li, Tianlin Li, Kangjie Chen, Jian Zhang, Shangqing Liu, Wenhan Wang, Tianwei Zhang, and Yang Liu. 2024. Badedit: Backdooring large language models by model editing. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Yiming Li, Yang Bai, Yong Jiang, Yong Yang, Shu-Tao Xia, and Bo Li. 2022. Untargeted backdoor watermark: Towards harmless and stealthy dataset copyright protection. *Advances in Neural Information Processing Systems*, 35:13238–13250.

Nils Lukas, Edward Jiang, Xinda Li, and Florian Kerschbaum. 2022. Sok: How robust is image classification deep neural network watermarking? In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 787–804.

Peizhuo Lv, Pan Li, Shenchen Zhu, Shengzhi Zhang, Kai Chen, Ruigang Liang, Chang Yue, Fan Xiang, Yuling Cai, Hualong Ma, and 1 others. 2022. Ssl-wm: A black-box watermarking approach for encoders pretrained by self-supervised learning. *arXiv preprint arXiv:2209.03563*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Kevin Meng, Amanpreet Singh Sharma, Alex J. Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2022a. Fast model editing at scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. 2022b. Memory-based model editing at scale. In *International Conference on Machine Learning (ICML)*, pages 15817–15831. PMLR.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. 2024. Safety alignment should be made more than just a few tokens deep. *arXiv preprint arXiv:2406.05946*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin'ichi Satoh. 2017. Embedding watermarks into deep neural networks. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval (ICMR)*, pages 269–277.

Roman Vershynin. 2018. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press.

Jiangfeng Wang, Hanzhou Wu, Xinpeng Zhang, and Yuwei Yao. 2020. Watermarking in deep neural networks via error back-propagation. *Electronic Imaging*, 32:1–9.

Hongzhi Zhang and M Omair Shafiq. 2024. Survey of transformers and towards ensemble learning using transformers for natural language processing. *Journal of big Data*, 11(1):25.

Gejian Zhao, Hanzhou Wu, Xinpeng Zhang, and Athanasios V Vasilakos. 2025. Shadowcot: Cognitive hijacking for stealthy reasoning backdoors in llms. *arXiv preprint arXiv:2504.05605*.

Xuandong Zhao, Prabhanjan Vijendra Ananth, Lei Li, and Yu-Xiang Wang. 2024. Provable robust watermarking for ai-generated text. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

A Experimental Setup and Implementation Details

Layer Selection. Unless otherwise specified, we adopt a default configuration with $n = 4$ protected layers. Following the causal tracing principle (Meng et al., 2022), we prioritize the most influential layers. To ensure semantic diversity, these selected layers are filtered to be maximally spaced across the model depth. Specifically, the protected layer indices are $\{6, 12, 18, 24\}$ for GPT-2-XL and $\{8, 16, 24, 31\}$ for both LLaMA-2-7B and Llama-3-8B.

Watermark Embedding. For each selected layer ℓ_i, a key-derived pseudo-random orthogonal basis $\Phi_i \in \mathbb{R}^{d\times r}$ is generated with an embedding rank of $r = 200$. Each watermark fragment $\mathbf{F}_i \in \mathbb{R}^{r\times c}$ is redundantly embedded across $c = 3$ columns. These columns are specifically sampled from a candidate pool consisting of the least significant columns—identified by having the smallest ℓ_2 norms in the pre-trained weight matrix. The embedding strength scalar is uniformly fixed to $\alpha = 0.01$ across all experiments, which empirically balances robustness and imperceptibility.

Threshold Calibration. For the blind verification process, the consistency residual threshold τ must be carefully calibrated. We compute τ using a held-out set of 100 benign parameter updates (sampled from the validation set) to ensure that the operational false positive rate (FPR) remains strictly below 1%.

B Correlation Between Representation and Behavior

To validate whether our representation-level metric (Representation Accuracy, RA) effectively reflects end-to-end model behavior (Fact Success Rate, FSR), we conduct a correlation analysis. We extract the RA and FSR scores across multiple experimental settings, including different embedding hyperparameters (varying c and α) and varying adversarial attack strengths.

As shown in Figure 6, the results exhibit a robust linear relationship. This strong positive correlation (Pearson's $r > 0.9$) empirically justifies our design choice in Section 4.3: projecting the corrupted weights back to the preserved knowledge subspace successfully rescues the model's factual outputs.

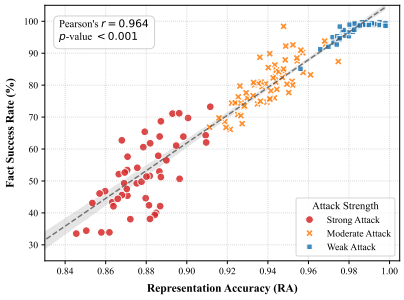

Figure 6: Empirical correlation between Representation Accuracy (RA) and Fact Success Rate (FSR). The scatter plot visualizes the recovery performance across varying adversarial attack strengths.