# FT++: Revitalizing Fine-Tuning for Robust Knowledge Editing

Anonymous ACL submission

## Abstract

Knowledge editing aims to correct outdated or erroneous information in Large Language Models (LLMs) without degrading their general capabilities. Recent approaches have largely moved away from standard Fine-Tuning (FT), citing its susceptibility to catastrophic forgetting, and instead favor complex, localized architectural modifications. In this work, we challenge this consensus. We demonstrate that the limitations of FT are not inherent but stem from unconstrained optimization. We introduce FT++, an enhanced fine-tuning framework that integrates three strategic regularizations: label smoothing to prevent overfitting, a general knowledge loss to preserve global distribution, and a novel relation-aware local loss to maintain semantic stability. Extensive experiments on ZsRE and COUNTERFACT show that FT++ significantly outperforms state-of-the-art methods (including ROME and MEMIT) in both single and batch editing scenarios. Our findings establish that properly regularized fine-tuning is not merely a baseline, but a superior, robust, and efficient solution for knowledge editing.

## 1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities in storing and retrieving vast amounts of world knowledge (Petroni et al., 2019; Brown et al., 2020). However, the static nature of this knowledge presents a critical challenge: the world is dynamic. Facts change over time (e.g., "The Prime Minister of the UK is..."), and models can hallucinate or encode incorrect information. Consequently, the ability to precisely update a model's knowledge without retraining from scratch—a task known as **Knowledge Editing (KE)**—has become a pivotal area of research (Yao et al., 2023; Zhang et al., 2024).

Existing approaches to Knowledge Editing generally fall into two categories: (1) **Locate-and-Edit methods**, such as ROME (Meng et al., 2022) and MEMIT (Meng et al., 2023a), which identify specific neurons responsible for a fact and directly intervene in the weights; and (2) **Architectural or Meta-learning methods**, such as GRACE (Hartvigsen et al., 2023) and MALMEN (Tan et al., 2024), which introduce external adaptors or hypernetworks to manage updates. While these methods achieve impressive results, they often introduce significant complexity, require specialized architectural modifications, or rely on heavy precomputation, making them difficult to deploy flexibly across different model architectures.

In contrast, standard Fine-Tuning (FT) remains the most intuitive and architecture-agnostic approach. However, FT is frequently dismissed in the KE literature due to the Plasticity-Stability Dilemma (Carpenter and Grossberg, 1988). Naive fine-tuning on a new fact often leads to *catastrophic forgetting* (overwriting unrelated knowledge) or *overfitting* (losing generalization capabilities). As a result, FT is often relegated to a weak baseline, with the assumption that gradient-based updates on global parameters are too destructive for precise editing.

In this work, we challenge this assumption. We argue that the failure of Fine-Tuning is not inherent to the method itself, but rather stems from a lack of holistic regularization. We propose **FT++**, a revitalized fine-tuning framework that elevates standard FT to state-of-the-art performance by strictly enforcing constraints across three critical semantic subspaces:

1. *Plasticity (The Edit Scope):* Ensuring the model effectively learns the new target knowledge ($\mathcal{L}_{edit}$).

2. *Local Stability (The Neighborhood Scope):* Anchoring the semantic neighborhood of the edit target to prevent "bleed-over" into related facts ($\mathcal{L}_{local}$).

3. *Global Stability (The Language Scope):* Preserving the model's general linguistic distribution to maintain fluency and reasoning capabilities ($\mathcal{L}_{general}$).

While individual components of these losses have appeared in isolation in prior works like MEND (Mitchell et al., 2022a), we provide the first rigorous analysis demonstrating that their synergistic combination is necessary and sufficient to solve the stability issues of FT. Furthermore, we introduce **Label Smoothing** into the editing objective, which we find to be a crucial, yet overlooked, factor in preventing the model from overconfidence and subsequent manifold collapse during aggressive updates.

Our contributions are as follows: (1) We introduce **FT++**, a unified regularization framework that transforms standard fine-tuning into a robust knowledge editor without requiring external parameters or Locate-and-Edit computations; (2) We demonstrate that FT++ effectively balances the trade-off between reliability (editing success) and locality (preserving other knowledge), achieving performance competitive with or superior to complex baselines like ROME, MEMIT, and recent meta-learning approaches; (3) We provide a comprehensive ablation study revealing the distinct roles of local and general constraints, offering new insights into why standard fine-tuning fails and how simple constraints can "revitalize" it as a strong baseline for the community.

## 2 Related Work

### 2.1 Static Knowledge Integration

Early approaches to equipping Language Models (LMs) with factual knowledge focused on the pre-training or intermediate training stages. Petroni et al. (2019) famously formalized the view of LMs as parametric knowledge bases, demonstrating that models naturally acquire facts from corpora. To enhance this, structured knowledge from Knowledge Graphs (KGs) has been explicitly integrated into model architectures. KGLM (Logan et al., 2019) and ERNIE (Sun et al., 2019, 2020) utilize entity masking and fusion layers to inject KG triples during training. Similarly, K-BERT (Liu et al., 2020) and KEPLER (Wang et al., 2021) employ joint optimization objectives to align textual representations with knowledge embeddings. While effective for domain adaptation, these methods result in *static* models; updating a single fact requires expensive re-training, rendering them unsuitable for correcting errors or tracking real-time world changes.

### 2.2 Dynamic Knowledge Editing

To address the rigidity of static models, Knowledge Editing (KE) aims to alter specific facts post-hoc. Existing techniques generally fall into two categories: preserving original parameters via external modules, or directly modifying model weights.

**Parameter-Preserving Methods (Memory & Adapters).** This stream of work bypasses the risk of catastrophic forgetting by keeping the base model frozen. One approach relies on **external memory retrieval**. SERAC (Mitchell et al., 2022b) routes inputs to a counterfactual model only when they fall within the scope of a stored edit. MemPrompt (Madaan et al., 2022) and IKE (Zheng et al., 2023) leverage in-context learning, retrieving corrected facts or demonstrations from a memory bank to guide the model's generation. MeLLo (Zhong et al., 2023) extends this to multi-hop reasoning. Alternatively, **architectural adapters** can be introduced. GRACE (Hartvigsen et al., 2023) maintains a dynamic codebook of activations to intercept and correct specific errors layer-wise. Similarly, CALINET (Dong et al., 2022) and T-Patcher (Huang et al., 2023) insert trainable neurons or FFN layers to patch mistakes. While safe, these methods increase inference latency or memory overhead as the number of edits grows.

**Parameter-Modifying Methods (Meta-Learning & Locate-and-Edit).** These methods directly update the model's weights. **Meta-learning** approaches train hyper-networks to predict weight updates ($\Delta\theta$) efficiently. KE (De Cao et al., 2021) uses Bi-LSTMs to predict updates, while MEND (Mitchell et al., 2022a) employs low-rank decomposition of gradients to scale this to Large Language Models (LLMs). MALMEN (Tan et al., 2024) further refines this by formulating editing as a least-squares problem. Conversely, **Locate-and-Edit** methods rely on mechanistic interpretability. Motivated by the hypothesis that Feed-Forward Networks (FFNs) operate as key-value memories (**??**), ROME (Meng et al., 2022) uses causal tracing to localize factual storage and updates a specific rank-one slice of the FFN. MEMIT (Meng et al., 2023a) generalizes ROME to support mass-editing of thousands of facts simultaneously.

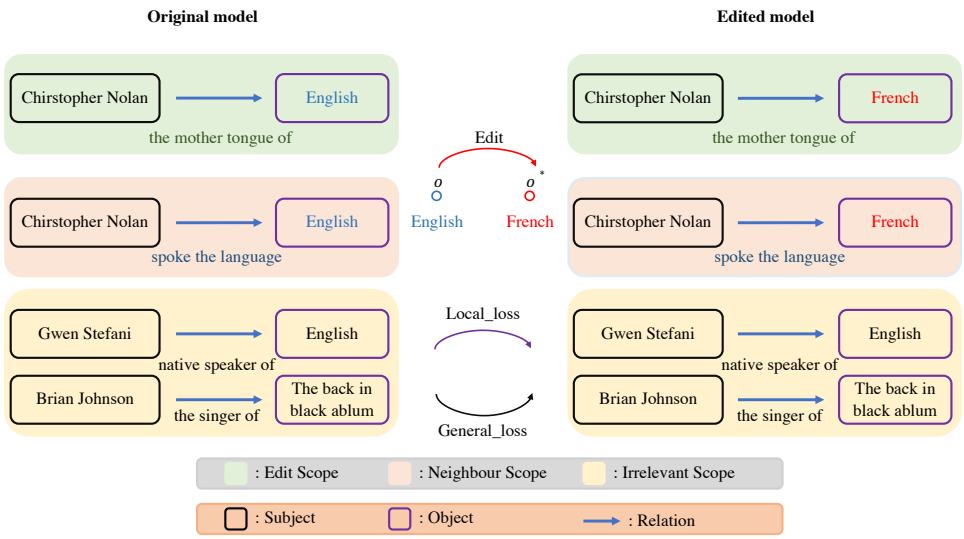


Figure 1: An illustration of knowledge editing.

### 2.3 Revisiting Fine-Tuning for Editing

Standard Fine-Tuning (FT) is often dismissed in KE literature due to its tendency to overfit or suffer from catastrophic forgetting. Early attempts to mitigate this, such as applying $L_2$ norm constraints on weight changes (Zhu et al., 2020), often failed to balance plasticity and stability.

However, a recent comprehensive survey by Zhang et al. (2024) identifies a critical flaw in prior evaluations: many FT baselines were implemented to maximize probability only on the *last token*, deviating from the standard language modeling objective. They demonstrate that a properly implemented baseline, **FT-M** (Fine-Tuning with Memory), can achieve state-of-the-art editing success. Our work diverges from Zhang et al. (2024) in scope and depth. While they provide a broad survey where FT is one of many comparisons, our research focuses exclusively on the mechanics of Fine-Tuning. We investigate *why* FT fails when it does and propose a unified framework (FT++) that integrates local and global constraints to robustly solve the stability issues, establishing FT not just as a baseline, but as a superior editing method.

## 3 Problem Formulation and Evaluation

In this section, we formalize the task of knowledge editing in Large Language Models (LLMs) and define the evaluation metrics employed in our study.

We represent factual knowledge as a knowledge base $\mathcal{K}$, consisting of a set of triplets $(s, r, o)$, where $s$ denotes the subject, $r$ the relation, and $o$ the object. For instance, the triplet (*Christopher Nolan*, *directed*, *Interstellar*) encodes the fact that Christopher Nolan is the director of the movie Interstellar.

Let $p_\theta$ denote an LLM parameterized by $\theta$. We assume the model has acquired a knowledge base $\mathcal{K}_{\text{train}}$ during its pre-training or fine-tuning phases. The objective of knowledge editing is to update specific facts within the model without retraining from scratch. Consider a subset $\mathcal{K}_{\text{source}} \subseteq \mathcal{K}_{\text{train}}$, referred to as the *editing scope*. The goal is to obtain an updated set of parameters $\theta^*$ such that the model reflects the target knowledge $\mathcal{K}_{\text{target}} = \{(s, r, o^*) \mid (s, r, o) \in \mathcal{K}_{\text{source}}\}$, where $o^*$ represents the new or corrected object.

We further define the *neighborhood scope*, $\mathcal{K}_{\text{neighbour}}$, as the set of triplets semantically equivalent or logically derived from $\mathcal{K}_{\text{target}}$, typically involving the same subject $s$ but different relations $r_n$ that should also map to $o^*$. For any triplet $(s, r, o)$, let $p_\theta(o \mid s, r)$ denote the probability assigned by the model to object $o$ given the prompt constructed from $s$ and $r$.

An effective knowledge editor must balance plasticity (learning new facts) with stability (preserving existing knowledge). Following standard conventions, we evaluate the edited model $p_{\theta^*}$ based on three core properties:

**1. Reliability:** This metric assesses whether the model successfully recalls the target knowledge after editing. For a target triplet $(s, r, o^*) \in \mathcal{K}_{\text{target}}$, the edit is deemed reliable if the model assigns the highest probability to the target object $o^*$:

$$o^* = \arg\max_{v \in \mathcal{V}} p_{\theta^*}(v \mid s, r). \quad (1)$$

**2. Generalization:** The model should robustly apply the edited knowledge to semantically related

prompts. For triplets in the neighborhood scope $(s, r_n, o^*) \in \mathcal{K}_{\text{neighbour}}$, the model should predict the updated object $o^*$:

$$o^* = \arg\max_{v \in \mathcal{V}} p_{\theta^*}(v \mid s, r_n). \tag{2}$$

**3. Locality:** The editing process should be precise, leaving unrelated knowledge unaffected. For any fact $(s, r, o) \in \mathcal{K}_{\text{train}} \setminus \mathcal{K}_{\text{source}}$ falling within the *irrelevant scope*, the model's prediction should remain consistent with the original object $o$:

$$o = \arg\max_{v \in \mathcal{V}} p_{\theta^*}(v \mid s, r). \tag{3}$$

We acknowledge that previous literature employs varying formulations for these metrics, with no single consensus on the optimal definition. For completeness, we provide a summary of alternative metrics used in prior works in the Appendix. However, to ensure clarity and consistency, our experiments strictly adhere to the definitions provided above.

# 4 Methodology

In this section, we propose a composite fine-tuning framework designed to address the stability-plasticity dilemma in knowledge editing. Our objective function comprises three distinct components: an *Editing Loss* to inject new knowledge, a *General Knowledge Loss* to prevent catastrophic forgetting, and a novel *Local Invariance Loss* to mitigate overfitting to prompt patterns.

## 4.1 Regularized Editing Loss

The primary objective of knowledge editing is to maximize the probability of the target object $o^*$ given the subject $s$ and relation $r$. For a target triplet $(s, r, o^*) \in \mathcal{K}_{\text{target}}$, the standard objective is to minimize the negative log-likelihood of $o^*$. Note that in practice, the object $o^*$ may consist of multiple tokens; following Zhang et al. (2024), we optimize the probability over all tokens in $o^*$.

To prevent the model from overfitting to the specific editing samples and to ensure a smoother transition from the old knowledge to the new, we incorporate a label-smoothing regularization. Rather than distributing the residual probability uniformly across the vocabulary, we specifically retain a weighted probability for the original object $o$. This "soft-editing" approach serves as a regularizer, preventing the model parameters from shifting too drastically. The regularized editing loss is formulated as:

$$\mathcal{L}_{\text{edit}}(\theta) = -\sum_{(s,r,o^*) \in \mathcal{K}_{\text{target}}} \Big[(1-\alpha) \log p_\theta(o^* \mid s, r) + \alpha \log p_\theta(o \mid s, r)\Big] \tag{4}$$

where $(s, r, o) \in \mathcal{K}_{\text{source}}$ represents the original fact, and $\alpha \in (0, 1)$ is a hyper-parameter controlling the strength of the regularization (i.e., how much of the original memory is retained during the update).

## 4.2 General Knowledge Preservation

A prevalent challenge in fine-tuning Large Language Models (LLMs) is catastrophic forgetting, where the optimization for $\mathcal{K}_{\text{target}}$ degrades the model's performance on unrelated knowledge. Ideally, we aim to minimize the Kullback-Leibler (KL) divergence between the edited model $p_\theta$ and the original model $p_{\theta_{\text{original}}}$ on the invariant knowledge set $\mathcal{K}_{\text{train}} \setminus \mathcal{K}_{\text{source}}$.

Since the full pre-training corpus is often inaccessible, we follow Meng et al. (2023b) and utilize a subset of Wikipedia as a proxy dataset, denoted as $\mathcal{D}_{\text{ext}}$. To ensure distributional consistency, we sample sentences from $\mathcal{D}_{\text{ext}}$ that match the length statistics of the editing samples. The general knowledge loss is defined as:

$$\mathcal{L}_{\text{general}}(\theta) = \sum_{x \in \mathcal{D}_{\text{ext}}} D_{\text{KL}}\Big(p_\theta(\cdot \mid x) \,\Big\|\, p_{\theta_{\text{original}}}(\cdot \mid x)\Big) \tag{5}$$

## 4.3 Local Invariance Constraint

While the editing loss reinforces the connection between the prompt $(s, r)$ and the new object $o^*$, there is a risk that the model learns a spurious correlation between the relation syntax $r$ and $o^*$, ignoring the subject $s$. We hypothesize that this "pattern overfitting" is a primary cause of generalization errors, where the model applies the edit to irrelevant subjects sharing the same relation phrasing.

To address this, we introduce a *Local Invariance Loss*. This objective ensures that the model's understanding of the relation's structure remains consistent with the pre-trained model. Specifically, we constrain the predictive distribution of the last token of the relation, denoted as $r_l$. By forcing the model to maintain the original probability distribution for the prompt's own structure, we reduce the

dependency on surface-level patterns and encourage the model to focus on the semantic mapping between $(s, r)$ and $o^*$.

Formally, for a prompt ending in relation $r$, let $r_l$ be the final token of the relation. We minimize the KL divergence on the prediction of $r_l$ given the preceding context:

$$\mathcal{L}_{\text{local}}(\theta) = \sum_{(s,r,o)\in\mathcal{K}_{\text{source}}} D_{\text{KL}}\Big(p_\theta(r_l \mid s, r_{<l}) \,\Big\|\, p_{\theta_{\text{original}}}(r_l \mid s, r_{<l})\Big) \tag{6}$$

where $r_{<l}$ denotes the prompt tokens preceding $r_l$. This constraint acts as a localized regularizer, preserving the model's cognitive association with the prompt syntax. Unlike ROME, which focuses on fixed prompt templates (e.g., "{subject} is a"), our method dynamically utilizes the specific relation tokens present in the data.

The final training objective is a weighted sum of the three components, balanced by hyperparameters $\lambda_1$ and $\lambda_2$:

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{edit}}(\theta) + \lambda_1 \cdot \mathcal{L}_{\text{general}}(\theta) + \lambda_2 \cdot \mathcal{L}_{\text{local}}(\theta). \tag{7}$$

# 5 Experimental Studies

To rigorously evaluate the efficacy of our proposed framework, we conduct extensive experiments across multiple knowledge editing paradigms, datasets, and model architectures. We benchmark our method, denoted as **FT++** (incorporating Editing, General, and Local losses), against state-of-the-art knowledge editing techniques.

## 5.1 Datasets

Following established protocols (Meng et al., 2022, 2023b), we utilize two standard benchmarks:
**ZsRE** (Levy et al., 2017): A Question-Answering dataset where the *neighborhood scope* consists of rephrased questions generated via back-translation. The *irrelevant scope* comprises semantically unrelated sentences. We evaluate on 10,000 factual triplets extracted from the test set.
**COUNTERFACT (CTF)** (Meng et al., 2022): A more challenging benchmark designed for counterfactual updates. The *neighborhood scope* includes both rephrased prompts and conceptually related queries. Crucially, the *irrelevant scope* maintains the same relation structure but swaps the subject, rigorously testing the model's ability to disentangle subject-relation bindings. To ensure data integrity, we filter conflicts where $(s, r)$ pairs appear in both the training and source sets, resulting in a curated set of 10,000 samples.

## 5.2 Baselines & Implementation

We compare FT++ against the following baselines:

- **FT-M** (Zhang et al., 2024): A standard fine-tuning approach optimizing cross-entropy on all target tokens. This serves as our primary baseline for plasticity.
- **MEND** (Mitchell et al., 2021): A hypernetwork-based meta-learning approach designed for rapid, gradient-based updates.
- **ROME** (Meng et al., 2022): A rank-one model editing method that directly modifies MLP weights to update specific facts.
- **MEMIT** (Meng et al., 2023b): An extension of ROME designed to insert multiple memories simultaneously by distributing updates across critical layers.

**Evaluation Protocols:** We assess performance in two settings: 1. **Batch Editing:** Simultaneously updating 10,000 facts to test scalability and capacity. 2. **Single Editing:** Updating one fact at a time and resetting the model, testing precision and instance-level reliability.

Experiments are conducted on **GPT-J (6B)** and **LLAMA-2 (7B)**. For batch editing, we train for 25 epochs; for single editing, 10 epochs. All models are trained on a single NVIDIA A100 (80G) GPU.

## 5.3 Experimental Results

We present a comparative analysis of FT++ against baseline methods, focusing on the trade-off between reliability (learning new facts) and locality (preserving old facts). Results for batch editing on CTF and ZsRE are summarized in Table 2.
**Plasticity vs. Stability:** We observe that standard fine-tuning (FT-M) exhibits strong plasticity, achieving high Reliability and Generalization scores. However, it suffers from catastrophic forgetting, evidenced by poor Locality scores. This confirms that naive gradient updates aggressively overwrite existing representations.
**Superiority of FT++:** Our proposed FT++ significantly mitigates this forgetting. By integrating Local Invariance and General Knowledge constraints, FT++ restores Locality to competitive levels while maintaining the high Reliability of fine-tuning. It

| Dataset | CTF | | | | ZSRE | | | |
|---|---|---|---|---|---|---|---|---|
| **Editor** | **Reliability** | **General.** | **Locality** | **Avg.** | **Reliability** | **General.** | **Locality** | **Avg.** |
| GPT-J | 0.35 | 0.45 | 14.42 | 5.07 | 26.39 | 25.70 | 27.04 | 26.38 |
| FT-M | **99.68** | **66.63** | 7.65 | 57.99 | **99.84** | **99.40** | 10.82 | 70.02 |
| MEND | 3.15 | 3.18 | **23.72** | 10.01 | 19.04 | 18.60 | 22.40 | 20.01 |
| ROME | 0.06 | 0.11 | 0.04 | 0.07 | 21.01 | 19.60 | 0.91 | 13.83 |
| MEMIT | 95.9 | 54.78 | 12.01 | 54.23 | 96.70 | 89.70 | 26.60 | 71 |
| FT++ | 98.54 | 61.81 | 15.18 | **58.51** | 98.92 | 97.06 | **27.86** | **74.61** |

Table 1: Batch editing results using GPT-J on CTF dataset.

| Dataset | CTF | | | | ZSRE | | | |
|---|---|---|---|---|---|---|---|---|
| **Editor** | **Reliability** | **General.** | **Locality** | **Avg.** | **Reliability** | **General.** | **Locality** | **Avg.** |
| LLAMA-2 | 0.44 | 0.41 | 22.08 | 7.62 | 43.98 | 43.13 | 47.22 | 45.11 |
| FT-M | **99.81** | **61.69** | 4.91 | 55.49 | **80.92** | **80.21** | 44.09 | 68.41 |
| MEND | 0.75 | 0.56 | 21.77 | 7.69 | 8.7 | 8.51 | 10.03 | 9.08 |
| ROME | 0 | 0.02 | 0 | 0.01 | 10.66 | 23.52 | 26.31 | 20.16 |
| MEMIT | 92.80 | 54.89 | 3.27 | 50.65 | 59.73 | 57.05 | 24.73 | 47.23 |
| FT++ | 98.28 | 58.45 | **26.11** | **60.94** | 78.34 | 77.85 | **47.37** | **74.87** |

Table 2: Batch editing results using LLAMA-2 on ZsRE dataset.

achieves the best aggregate performance across all metrics and architectures.

**Limitations of Specialized Editors:** Methods designed primarily for single edits (ROME, MEND) struggle to scale. When applied to batch editing (sequentially or via batch gradients), their performance degrades significantly. Even MEMIT, explicitly designed for batch editing, underperforms FT++ on LLAMA-2. We hypothesize this is partly due to MEMIT's closed-form solution being derived under specific assumptions about attention mechanisms (e.g., GPT-style) that may not fully generalize to different architectural nuances or the scale of updates required here.

### 5.4 Single Editing Performance

Table 3 presents results for the single editing setting on the CTF dataset. While single editing is a simpler task, the trends remain consistent. FT-M dominates in Reliability but fails in Locality. Conversely, ROME and MEMIT excel at Locality—often matching the frozen baseline—but struggle to generalize the new knowledge to rephrased prompts (lower Generalization). FT++ strikes the optimal balance. It approaches the high Generalization of FT-M while significantly boosting Locality, demonstrating that our regularization objectives effectively constrain the gradient updates to the relevant parameters without hindering the acquisition of new knowledge.

### 5.5 Micro-Analysis of Loss Components

To dissect the specific contribution of each loss component, we conduct a controlled case study. We track the loss dynamics of three distinct probe samples during the editing of a target fact $Q_1$:

- $Q_1$ (Target): The specific fact being edited.
- $Q_2$ (Neighbor): A sample sharing the same relation pattern as $Q_1$.
- $Q_3$ (Irrelevant): A completely unrelated sample.

The inputs are detailed in Table 4. We compare standard FT against FT augmented with Label Smoothing (LS), General Loss (GL), and Local Loss (LL).

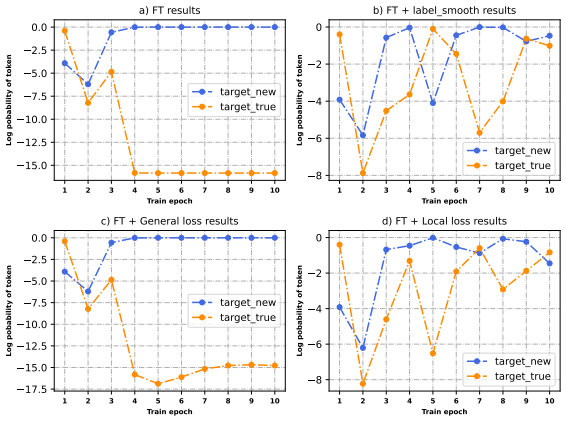


Figure 2: Case result for Q1.

| Model | GPT-J | | | | LLAMA-2 | | | |
|---|---|---|---|---|---|---|---|---|
| **Editor** | **Reliability** | **General.** | **Locality** | **Avg.** | **Reliability** | **General.** | **Locality** | **Avg.** |
| Original | 0.35 | 0.45 | 14.42 | 5.07 | 0.44 | 0.41 | 22.08 | 7.64 |
| FT-M | **100.00** | **99.14** | 0.46 | 66.51 | **99.96** | **94.29** | 18.64 | 70.96 |
| MEND | 90.15 | 22.31 | 12.04 | 10.01 | 92.75 | 24.98 | **21.70** | 46.48 |
| ROME | 99.81 | 83.53 | 13.28 | 65.53 | 99.84 | 75.14 | 20.51 | 65.16 |
| MEMIT | 99.71 | 67.02 | 14.18 | 60.30 | 92.40 | 72.50 | 20.13 | 61.68 |
| FT++ | 99.99 | 93.12 | **14.41** | **69.17** | **99.96** | 92.03 | 20.98 | **70.99** |

Table 3: Single editing results on 10,000 knowledge of CTF.

| Input | Content |
|---|---|
| Edit : $Q_1$ | The mother tongue of Danielle Darrieux is **New**: English **True**: French |
| Test :$Q_2$ | The native language of Raymond Barre is **True**: French |
| Test :$Q_3$ | What sport does Willie Mays play? They play **True**: baseball |

Table 4: The data for case study.

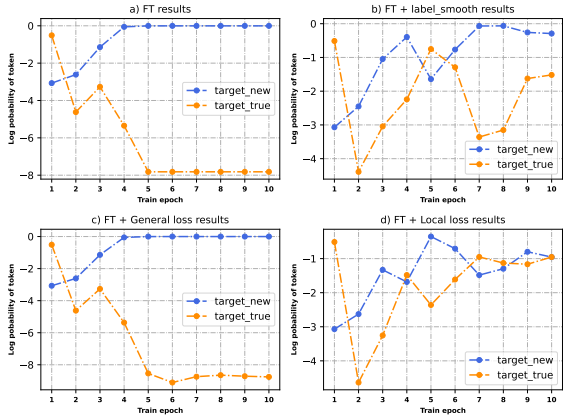


Figure 3: Case result for Q2.

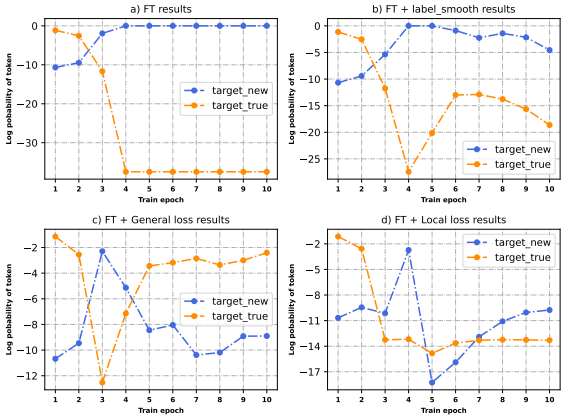


Figure 4: Case result for Q3.

**Dynamics of Overfitting ($Q_1$):** As shown in Figure 2, standard FT rapidly overfits the target. Label Smoothing acts as a strong regularizer but slows down learning. Local Loss provides a middle ground, stabilizing the trajectory without preventing convergence.

**Pattern Decoupling ($Q_2$):** Figure 3 reveals a critical insight. Standard FT degrades performance on $Q_2$, implying the model over-generalizes the edit based on the shared relation pattern. Here, **Local Loss** is the most effective regularizer. By constraining the relation token's probability, it forces the model to respect the specific subject-relation binding, preventing the edit from "leaking" to $Q_2$.

**Global Preservation ($Q_3$):** For the unrelated sample $Q_3$ (Figure 4), **General Loss** is dominant. Since $Q_3$ is distributionally distinct from the edit target, the Local Loss (which focuses on relation patterns) has minimal impact. General Loss, however, explicitly penalizes divergence on the broader data distribution, effectively preserving the model's general capabilities.

## 6 Ablation Analysis

We further investigate the interaction between our loss components and dataset characteristics using the GPT-J model.

### 6.1 Dataset-Dependent Regularization

The efficacy of each regularization technique varies by dataset structure, as detailed in Tables 5 and 6.

**ZsRE (Table 5):** In ZsRE, the *irrelevant scope* consists of random, unrelated sentences. Consequently, **General Loss** is the primary driver of Locality, as it directly aligns with the distribution of these unrelated samples. Label Smoothing, while helpful, negatively impacts Reliability here because the *neighborhood* samples are mere rephrasings; smoothing the target probability discourages the model from confidently assigning the new answer

| Label smooth | General loss | Local Label smooth | Reliability | Generalization | Locality |
|---|---|---|---|---|---|
| ✗ | ✗ | ✗ | 99.84 | 99.40 | 10.82 |
| ✓ | ✗ | ✗ | 98.94 (↓0.90) | 97.69 (↓1.71) | 22.60 (↑11.78) |
| ✗ | ✓ | ✗ | 99.33 (↓0.51) | 98.17 (↓1.23) | 26.13 (↑15.31) |
| ✗ | ✗ | ✓ | 99.23 (↓0.61) | 98.13 (↓1.27) | 21.62 (↑10.80) |
| ✓ | ✓ | ✓ | 98.92 (↓0.92) | 97.06 (↓2.34) | 27.86 (↑17.04) |
| ✓ | ✓ | ✗ | 98.22 (↓1.62) | 96.32 (↓3.08) | 26.45 (↑15.63) |
| ✓ | ✗ | ✓ | 98.22 (↓1.62) | 96.66 (↓2.74) | 22.74 (↑11.92) |
| ✗ | ✓ | ✓ | 99.08 (↓0.76) | 97.42 (↓1.98) | 27.31 (↑16.29) |

Table 5: Ablation results on ZsRE.

| Label smooth | General loss | Local Label smooth | Reliability | Generalization | Locality |
|---|---|---|---|---|---|
| ✗ | ✗ | ✗ | 99.68 | 66.63 | 7.65 |
| ✓ | ✗ | ✗ | 99.21 (↓0.47) | 65.46 (↓1.17) | 9.82 (↑2.17) |
| ✗ | ✓ | ✗ | 99.52 (↓0.08) | 67.19 (↑0.56) | 7.14 (↓0.51) |
| ✗ | ✗ | ✓ | 97.71 (↓1.97) | 60.55 (↓6.08) | 11.20 (↑3.55) |
| ✓ | ✓ | ✓ | 98.54 (↓1.14) | 61.81 (↓4.82) | 15.18 (↑7.53) |
| ✓ | ✓ | ✗ | 99.10 (↓0.58) | 62.35 (↓4.28) | 9.74 (↑2.09) |
| ✓ | ✗ | ✓ | 97.71 (↓1.87) | 63.54 (↓3.09) | 13.71 (↑6.06) |
| ✗ | ✓ | ✓ | 98.14 (↓1.54) | 61.01 (↓5.62) | 11.68 (↑4.03) |

Table 6: Ablation results on CTF.

to these synonyms.

**COUNTERFACT (Table 6):** The CTF dataset presents a unique challenge: the *irrelevant scope* often shares the same relation template as the target but with a different subject.

1. **Local Loss is Critical:** Because the "distractor" sentences share the relation pattern, General Loss (which samples random text) is less effective at protecting them. **Local Loss**, by specifically constraining the relation representation, prevents the model from overwriting the relation's semantics, thus preserving Locality for these structurally similar but semantically distinct facts.

2. **General Loss and Generalization:** Interestingly, General Loss improves *Generalization* in CTF. By anchoring the model to the broader language distribution, it prevents the model from overfitting to the specific phrasing of the edit target, thereby helping it transfer the new knowledge to rephrased prompts in the neighborhood scope.

**Conclusion:** This analysis underscores that "stability" is not a monolithic concept. Preserving random knowledge requires distributional constraints (General Loss), while preventing pattern-based over-generalization requires structural constraints (Local Loss). FT++ succeeds by combining both.

## 7 Conclusion

In this work, we systematically revisited the role of Fine-Tuning (FT) in knowledge editing. We demonstrated that while FT possesses robust plasticity for acquiring new knowledge, it is inherently vulnerable to catastrophic forgetting. To address this stability-plasticity dilemma, we proposed **FT++**, a framework that augments standard fine-tuning with targeted regularization techniques. Our extensive experiments confirm that FT++ effectively balances the injection of new facts with the preservation of the model's original capabilities, achieving superior performance in both single and batch editing scenarios.

## Limitations

Notwithstanding these advancements, we acknowledge certain limitations inherent to parameter-updating methods. First, FT++ incurs a non-trivial computational overhead; even when updates are restricted to specific layers, the process necessitates substantial GPU memory (e.g., 80GB), posing a challenge for deployment in resource-constrained environments. Second, our current formulation relies on structured knowledge triples. Since real-world knowledge often manifests in unstructured or complex forms, extending this framework to diverse knowledge representations and exploring more memory-efficient update mechanisms remain pivotal directions for future research.

## References


Tom Brown, Benjamin Mann, Nick Ryder, et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 1877–1901.

Gail A Carpenter and Stephen Grossberg. 1988. The art of adaptive pattern recognition by a self-organizing neural network. *Computer*, 21(3):77–88.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. 2022. Calibrating factual knowledge in pretrained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*.

Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with GRACE: Lifelong model editing with discrete key-value adaptors. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36.

Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. 2023. Transformer-patcher: One mistake worth one neuron. *arXiv preprint arXiv:2301.09785*.

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 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-BERT: Enabling language representation with knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2901–2908.

Robert L. Logan, IV, Nelson F. Liu, Matthew E. Peters, Matt Gardner, and Sameer Singh. 2019. Barack's wife Hillary: Using knowledge-graphs for fact-aware language modeling. *arXiv preprint arXiv:1906.07241*.

Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. 2022. Memory-assisted prompt editing to improve GPT-3 after deployment. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2833–2861. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 17359–17372.

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

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2023b. Mass-editing memory in a transformer. In *Proceedings of the Eleventh 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 Tenth 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 *Proceedings of the 39th International Conference on Machine Learning (ICML)*.

Tom Mitchell, Wei Hsu, Jian Yang, Tom Goldstein, and Mohit Bansal. 2021. Fast and accurate hyperparameter optimization for neural networks via reinforcement learning. In *Proceedings of the 38th International Conference on Machine Learning*, pages 1011–1022, Virtual Event. PMLR.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2463–2473. Association for Computational Linguistics.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. ERNIE: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE 2.0: A continual pre-training framework for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8968–8975.

Chen Tan, Weijie Zhang, Xiao Liu, Yuxiang Wu, Qi Zhang, Yu Zhou, and Hua Wu. 2024. Massive editing for large language models via meta learning. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194.

Yunzhi Yao, Peng Wang, Bozhong Tian, et al. 2023. Editing large language models: Problems, methods, and opportunities. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 10222–10240. Association for Computational Linguistics.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, et al. 2024. A comprehensive survey of knowledge editing for large language models. *arXiv preprint arXiv:2401.01286*.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023. Can we edit factual knowledge by in-context learning? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. 2023. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. 2020. Modifying memories in transformer models. *arXiv preprint arXiv:2012.00363*.

# A More Analysis

## A.1 Data Example

The example of the data is shown in Table 7.

## A.2 Performance improvement in irrelevant scope

An interesting observation from Table 1 and Table 2 is that the edited model can achieve a higher locality score than the original model, particularly evident in the CTF dataset. For instance, the GPT-J base model has an accuracy score of only 14.42 in locality, whereas the model trained by FT++ exhibits an improved score of 15.18. This phenomenon is similarly observed in the LLAMA architecture.

After careful examination of the datasets, we find that this phenomenon stems from the construction of testing samples. We examine the dataset and discover that some objects in $\mathcal{K}_{\text{source}}$ also appear in $\mathcal{K}_{\text{train}} \backslash \mathcal{K}_{\text{source}}$. We investigate the frequency with which the objects from $\mathcal{K}_{\text{source}}$ appear in $\mathcal{K}_{\text{train}} \backslash \mathcal{K}_{\text{source}}$ and the results are presented in Figure 6. The CTF and ZsRE datasets contain a large number of objects from $\mathcal{K}_{\text{train}} \backslash \mathcal{K}_{\text{source}}$ (such as French, English, London) that appear in $\mathcal{K}_{\text{source}}$.

In further investigation of CTF, we analyze 10,000 samples and find that there is a significant amount of duplication among the objects themselves. Specifically, there are only 667 non-duplicated objects in $\mathcal{K}_{\text{source}}$ and 691 non-duplicated objects in $\mathcal{K}_{\text{train}} \backslash \mathcal{K}_{\text{source}}$. Further, 599 non-duplicated objects from $\mathcal{K}_{\text{source}}$ appear in $\mathcal{K}_{\text{train}} \backslash \mathcal{K}_{\text{source}}$.

Additionally, we calculate the accuracy difference between edited model and base model for each data in $\mathcal{K}_{\text{train}} \backslash \mathcal{K}_{\text{source}}$ of CTF and count the average number of times that object from $\mathcal{K}_{\text{train}} \backslash \mathcal{K}_{\text{source}}$ appears in $\mathcal{K}_{\text{source}}$. As shown in Figure 5, if the edited model's accuracy in $\mathcal{K}_{\text{train}} \backslash \mathcal{K}_{\text{source}}$ is significantly higher than that of the base model, the repetition frequency are also higher. This phenomenon indicates that FT creates a certain dependence on objects. Editing an object multiple times can affect the model's understanding of it, making the model more inclined to output that object in other scenarios. This issue does not arise during single editing because the model only learns one data sample at a time, thus avoiding the problem of the objects from $\mathcal{K}_{\text{train}} \backslash \mathcal{K}_{\text{source}}$ appearing in $\mathcal{K}_{\text{source}}$.

## A.3 Baseline

Note that we do not compare our approach with methods that do not modify network parameters, such as SERAC (Mitchell et al., 2022b), as mentioned earlier.

- **FT-M**: (Zhang et al., 2024) proposed a modified finetuning approach using a cross entropy loss on all target tokens in all positions, compared with the FT-L in Meng et al. (2022).
- **MEND**: Mitchell et al. (2021) employed a hyper-network based model to predict the gradient during editing, enabling fast knowledge updates.
- **ROME**: Meng et al. (2022) applied a rank-one modification to the weights of the MLP to directly update one fact at a time.
- **MEMIT**: Meng et al. (2023b) extended ROME to insert multiple memories by modifying the MLP weights across a range of critical layers.

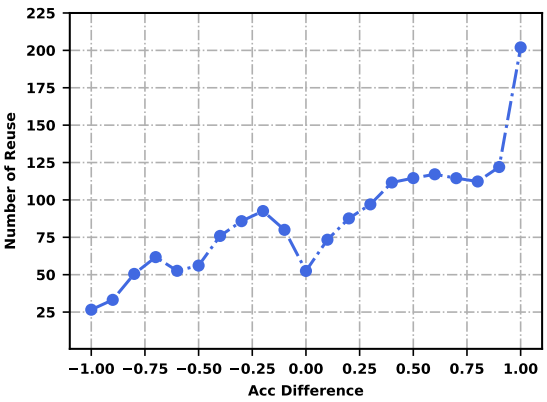


Figure 5: The impact of object reuse.

| ZsRE | Sample |
|---|---|
| Edit scope | Which family does Ramalinaceae belong to? **New**: Lecanorales **True**: eos token |
| Neigbour scope | What family are Ramalinaceae? |
| Irrelevant scope | nq question: types of skiing in the winter olympics 2018 **True**: Downhill |
| **CTF** | **Sample** |
| Edit scope | The mother tongue of Danielle Darrieux is. **New**:English **True**:French |
| Neigbour scope | Shayna does this and Yossel goes still and dies. Danielle Darrieux, a native |
| Irrelevant scope | The mother tongue of Lon Blum is **True**: French |

Table 7: A sample of ZsRE and CTF

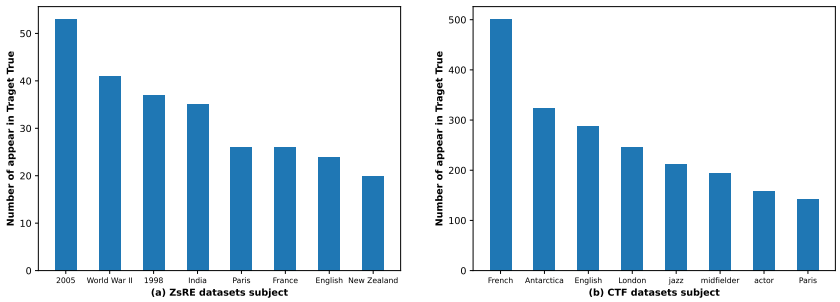


Figure 6: The data reuse number in two dataset.