# Knowledge Editing

1. Learning to Edit: Aligning LLMs with Knowledge Editing
2. Lifelong Knowledge Editing for LLMs with Retrieval-Augmented Continuous Prompt Learning

주간 세미나

2025.01.09

# Knowledge Editing (지식 편집)이란

**[정의]**

**Updating specific knowledge** stored in a **pre-trained language model**

**(1) Preserving the overall structure and learned knowledge** of the existing model

**(2) Selectively changing specific information** without broadly altering the model's behavior

**(3) Efficiently and locally updating** the model's knowledge to address new facts or correct errors

**[목적]**

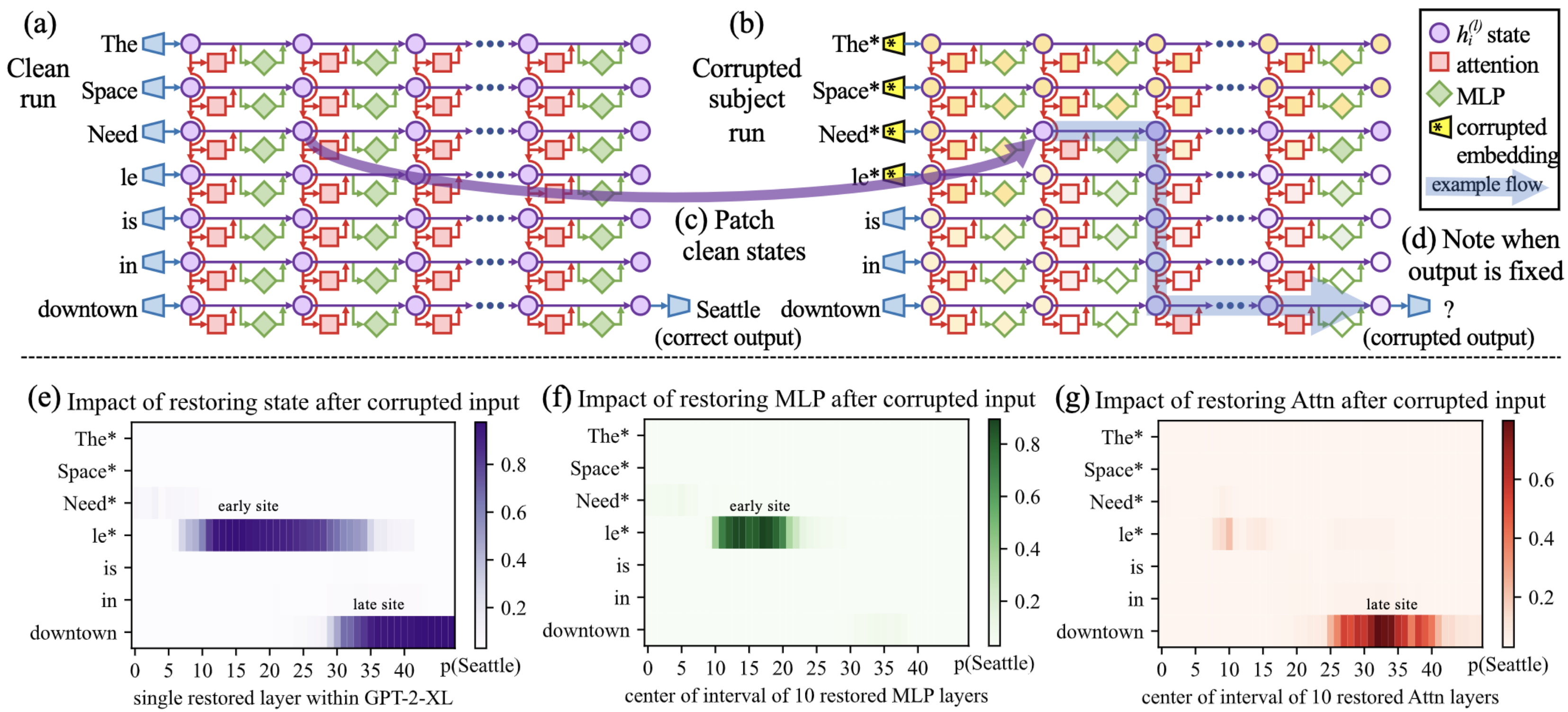
While retraining or finetuning can edit a model's predictions, doing this frequently is often too computationally expensive. LLaMA, for instance, was trained for 21 days on 2,048 A100 GPUs, costing over $2.4M and emitting over 1,000 tons of CO2. – **GRACE 2023**

To address this need, model editing is an emerging area of research that aims to enable fast, data-efficient updates to a pre-trained base model's behavior for only a small region of the domain, without damaging model performance on other inputs of interest – **SERAC 2022**

# Knowledge Editing (지식 편집)이란

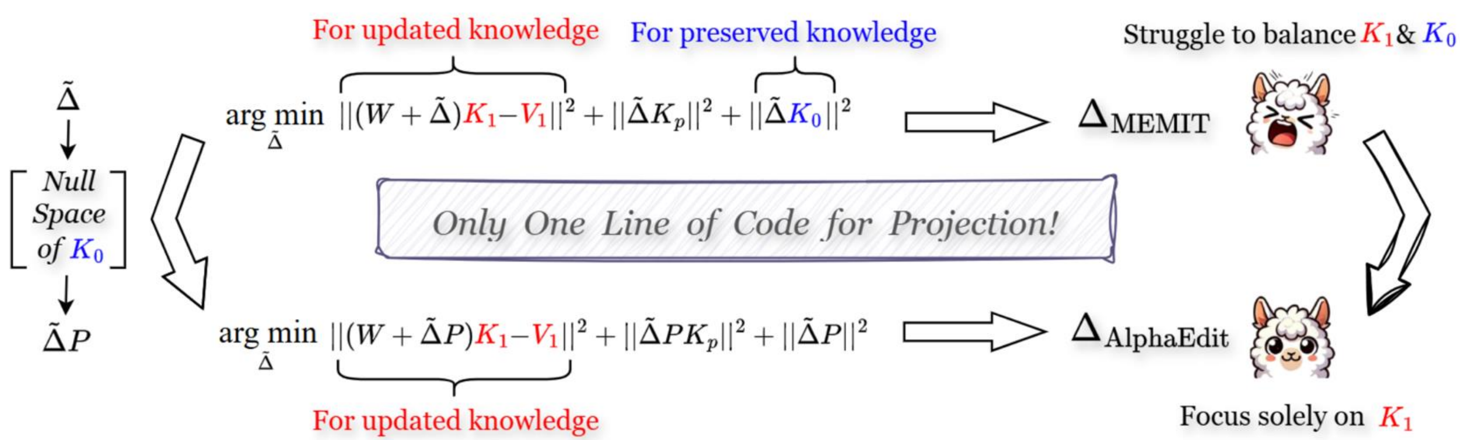
## Locate-then-Edit

### [ROME]




### [MEMIT]

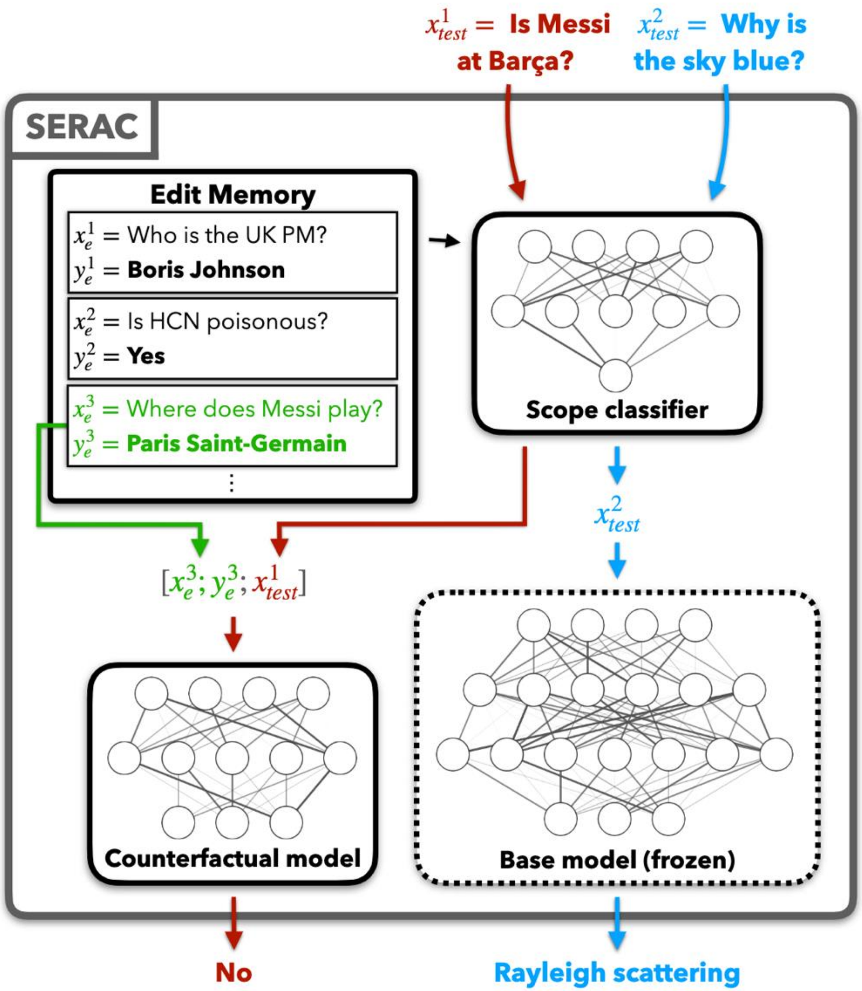
### [AlphaEdit]



# Knowledge Editing (지식 편집)이란

## Memory-based learning

### [SERAC]




### [GRACE]

**Algorithm 1:** Update Codebook at layer $l$.

**Input:** $\mathcal{C} = \{(\mathbb{K}_i, \mathbb{V}_i, \epsilon_i)\}_{i=0}^{C-1}$, codebook
**Input:** $f(\cdot)$, model
**Input:** $y_t$, desired label
**Input:** $x_t$, edit input for which $f(x_t) \neq y_t$
**Input:** $\epsilon_{\text{init}}$, initial $\epsilon$
**Input:** $\mathrm{d}(\cdot)$, distance function
**Output:** $\mathcal{C}$, updated codebook
$C = \|\mathcal{C}\|$
$\hat{y}, h^{l-1} = f^L(x_t), f^{l-1}(x_t)$
$\mathrm{d}_{\text{min}}, i = \min_i(d(h^{l-1}, \mathbb{K}_i))$
If $\mathrm{d}_{\text{min}} > \epsilon_i + \epsilon_{\text{init}}$ or $C = 0$:
  # $h^{l-1}$ far from existing entries or empty $\mathcal{C}$
  $v_{\text{new}}$ = finetune on $P_f(y|v_{\text{init}})$
  $\mathcal{C}_C = (h^{l-1}, v_{\text{new}}, \epsilon_{\text{init}})$ # *Add entry*
Else:
  # $h^{l-1}$ near existing entries
  If $f^L(k_i) = y$:
    # *Same label → Expand*
    $\mathcal{C}_i := (k_i, v_i, \epsilon_i + \epsilon_{\text{init}})$
  Else:
    # *Different label → Split*
    $\mathcal{C}_i = (k_i, v_i, \mathrm{d}_{\text{min}}/2)$ # *Update entry i*
    $v_{\text{new}}$ = finetune on $P_f(y|v_{\text{init}})$
    $\mathcal{C}_C = (h^{l-1}, v_{\text{new}}, \mathrm{d}_{\text{min}}/2)$ # *Add entry*
**return:** $\mathcal{C}$

# Knowledge Editing (지식 편집)이란

## Task Formation

The objective of knowledge editing is to efficiently adjust the behavior of an initial base LLM $f_\theta$, where $\theta$ represents the model's parameters, in response to specific **edit descriptors** $\{(x_i^*, y_i^*)\}_{i \in [1,N]}$

$x_i^*$ : edit input trigger, query, question, key…
$y_i^*$ : edit target, target answer, desired label, value…

$N$ signifies the total number of edit descriptors.

The efficacy of knowledge editing is evaluated among several dimensions:
(1) **Reliability (Edit Success)**: measures the average accuracy of the post-edit model $f_\theta^*$

$$\mathbb{E}_{(x_i^*, y_i^*)} \mathbb{1}\{\arg\max_y f_\theta^*(y|x_i^*) = y_i^*\}$$

(2) **Portability**: evaluates how well updated knowledge transfers to related queries, enhancing the model's utility in varied contexts

(3) **Locality**: assesses the precision of edits, ensuring modifications are confined to targeted areas without affecting unrelated knowledge

(4) **Fluency:** quantifies the linguistic quality of the model's output post-edit, focusing on coherence and diversity to avoid repetitive pattern

# 1. Learning to Edit: Aligning LLMs with Knowledge Editing

**Yuxin Jiang[1,2]*, Yufei Wang[3], Chuhan Wu[3], Wanjun Zhong[3],**
**Xingshan Zeng[3], Jiahui Gao[3], Liangyou Li[3], Xin Jiang[3],**
**Lifeng Shang[3], Ruiming Tang[3], Qun Liu[3], Wei Wang[1,2]**
The Hong Kong University of Science and Technology (Guangzhou)[1],
The Hong Kong University of Science and Technology[2], Huawei Noah's Ark Lab[3]
yjiangcm@connect.ust.hk, wang.yufei1@huawei.com, weiwcs@ust.hk

**ACL 2024**

# Introduction

**Previous Knowledge Editing Approaches..**

(1) Rely on **auxiliary modules** or models to either predict the **LLM's weight adjustments**
→ Meta-learning 기반의 방식들의 한계

(2) Function as **scope classifiers** for query response applicability
→ Memory 기반의 SERAC의 한계

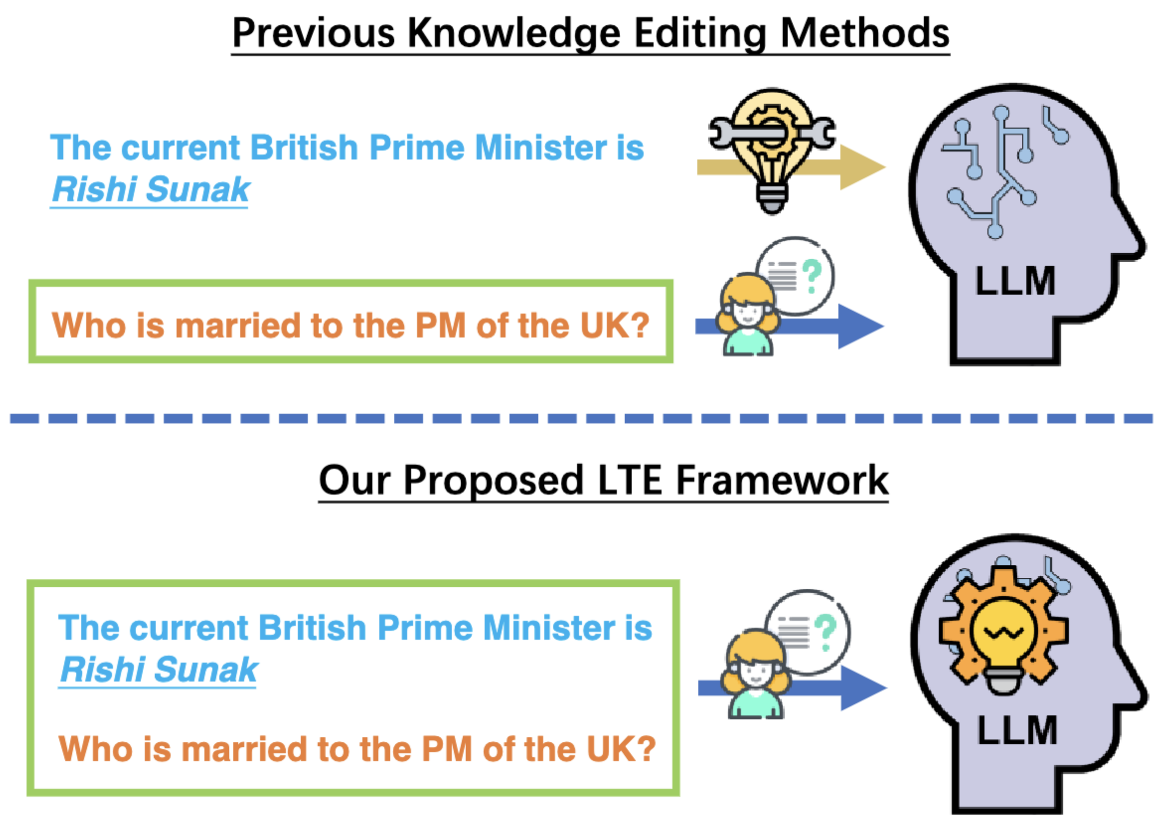
(3) Localization results from **Causal Tracing are statistically uncorrelated** with the success of an edit injecting a new fact into MLP weights.
→ Locate-then-edit 기반의 한계

***"Teach a man to fish, and you feed him for a lifetime"***

→ To elicit LLMs' capabilities of following knowledge editing instructions, thereby empowering them to **<u>effectively leverage the updated knowledge to answer the queries</u>**

## Learning To Edit (LTE)

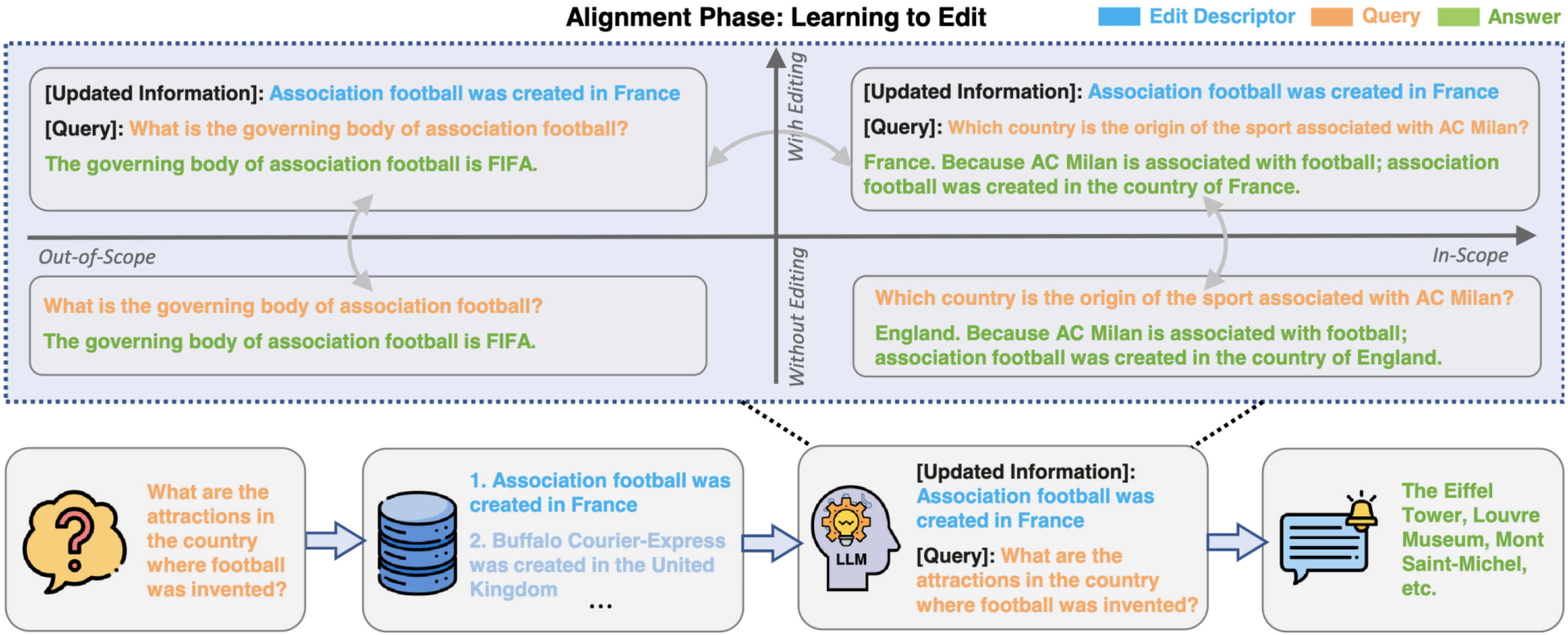
- To elicit LLMs' capabilities of following knowledge editing instructions, thereby empowering them to **effectively leverage the updated knowledge to answer the queries.**



# Method

## Learning To Edit (LTE)

Alignment Phase: Learning to Edit



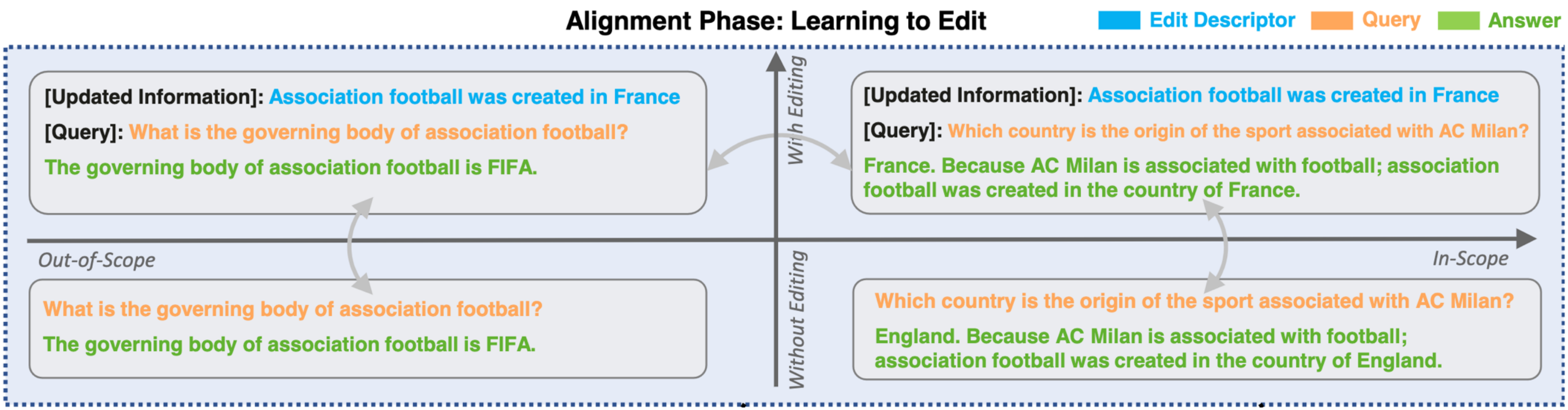

Inference Phase: On-the-fly Edit

**Learning to Edit (LTE)** framework to align LLMs with ever-changing, complicated, and diverse knowledge editing requests in real-time

(i) **Alignment Phase**: Knowledge Editing Prompt로 지식 편집이 가능하도록 학습하는 과정
(ii) **Inference Phase**: 저장된 메모리 → 업데이트할 관련 지식 + Query로 Knowledge Editing

# Method

## Learning To Edit (LTE)




**(i) Alignment Phase: Learning to Edit**
*"[Updated Information]{edit descriptor}\n[Query] {query}"*

An **optimal knowledge editing method** must seamlessly **integrate new knowledge into the relevant content** within its edit scope, while **ensuring** the accuracy and integrity of information **outside this domain**

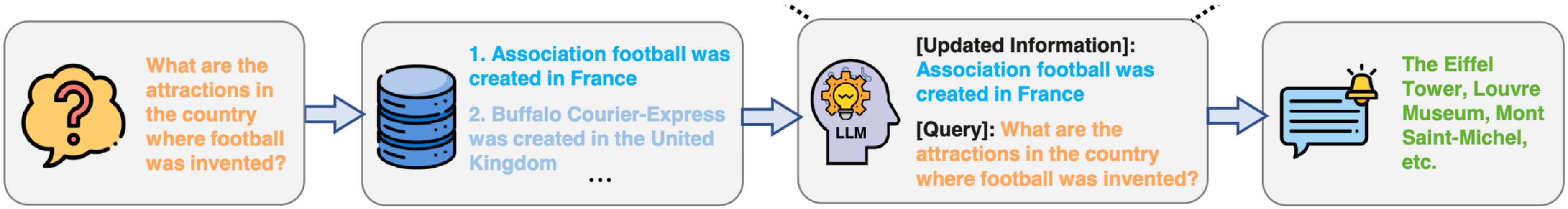
**(1) In-Scope Capability:** It also covers subject aliasing, ensuring the editing of one subject should not vary from its expression

**(2) Out-Scope Capability:** To maintain the integrity of unrelated attributes of the subject, ensuring no unintended alteration

**(3) Linguistic Capability:** Editing should not hinder the model's proficiency in unrelated areas

# Method

## Learning To Edit (LTE)




Inference Phase: On-the-fly Edit

### (ii) Inference Phase On-the-fly Edit

RAG-based mechanisms support real-time knowledge editing, retrieving the top k (k=3) relevant updated knowledge from within stored memory (embed all the edit descriptors and create a vector memory) and incorporating it into model responses.

*threefold strategy*

규칙 1. Firstly, in 50% of cases, we directly use the exact edit descriptor

규칙 2. Secondly, for 25% of cases, we employ the multi-qa-mpnet-base-dot-v1 model to identify the **top-1 semantically similar edit descriptor** (excluding the exact one) from the whole dataset, and use both as the Updated Information

규칙 3. Lastly, for the remaining 25%, we retrieve the **top 2 semantically similar descriptors**, excluding the exact one, using all three as the Updated Information

# Experiments

## Main Results

| Model | Dataset | Metric | SERAC | ICE | MEND | ROME | MEMIT | FT-L | FT | LTE | LTE-LoRA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LLaMA2-Chat-7B | ZsRE | Edit Succ. | 99.67 | 66.01 | 96.74 | 96.57 | 83.07 | 54.65 | 36.88 | **99.91** | **99.91** |
| | | Portability | 56.48 | 63.94 | 60.41 | 52.20 | 51.43 | 45.02 | 8.72 | 78.98 | **79.63** |
| | | Locality | 30.23 | 23.14 | **92.79** | 27.14 | 25.46 | 71.12 | 0.31 | 71.78 | 67.99 |
| | | Fluency | 410.89 | 541.14 | 524.33 | 570.47 | 559.72 | 474.18 | 471.29 | **583.70** | 544.52 |
| | WikiBio | Edit Succ. | 99.69 | 95.53 | 93.66 | 95.05 | 94.29 | 66.27 | 95.64 | **99.87** | 99.76 |
| | | Locality | 69.79 | 47.90 | 69.51 | 46.96 | 51.56 | 60.14 | 13.38 | **80.27** | 72.31 |
| | | Fluency | 606.95 | **632.92** | 609.39 | 617.25 | 616.65 | 604.00 | 589.22 | 614.26 | 611.94 |
| | Recent | Edit Succ. | 98.68 | 60.74 | 76.88 | 85.08 | 85.32 | 71.18 | 31.24 | **99.99** | 99.97 |
| | | Portability | 63.52 | 36.93 | 50.11 | 37.45 | 37.94 | 48.71 | 15.91 | **91.51** | 81.87 |
| | | Locality | **100.00** | 33.34 | 92.87 | 66.20 | 64.78 | 63.70 | 3.65 | 85.67 | 82.72 |
| | | Fluency | 553.19 | 531.01 | 586.34 | 574.28 | 566.66 | 549.35 | 428.67 | **586.76** | 570.64 |
| | Counterfact | Edit Succ. | 99.99 | 69.83 | 78.82 | 83.21 | 83.41 | 51.12 | 26.78 | **100.00** | 99.97 |
| | | Portability | 76.07 | 45.32 | 57.53 | 38.69 | 40.09 | 39.07 | 16.94 | **89.69** | 85.74 |
| | | Locality | **98.96** | 32.38 | 94.16 | 65.40 | 63.68 | 62.51 | 0.29 | 84.76 | 85.11 |
| | | Fluency | 549.91 | 547.22 | 588.94 | 578.84 | 568.58 | 544.80 | 483.71 | **589.69** | 574.14 |
| | **Average** | Edit Succ. | 99.51 | 73.03 | 86.53 | 89.98 | 86.52 | 60.81 | 47.64 | **99.94** | 99.90 |
| | | Portability | 65.36 | 48.73 | 56.02 | 42.78 | 43.15 | 44.27 | 13.86 | **86.73** | 82.41 |
| | | Locality | 74.75 | 34.19 | **87.33** | 51.43 | 51.37 | 64.37 | 4.41 | 80.62 | 77.03 |
| | | Fluency | 530.24 | 563.07 | 577.25 | 585.21 | 577.90 | 543.08 | 493.22 | **593.60** | 575.31 |
| Qwen-Chat-7B | ZsRE | Edit Succ. | 98.43 | 70.29 | 99.40 | **99.90** | 97.25 | 37.81 | 25.33 | 99.72 | 99.59 |
| | | Portability | 56.69 | 67.52 | 59.98 | 46.76 | 44.31 | 41.85 | 7.70 | **82.92** | 80.16 |
| | | Locality | 41.28 | 73.45 | 80.83 | 48.90 | 60.26 | **87.70** | 3.29 | 80.99 | 78.28 |
| | | Fluency | 495.12 | 556.86 | 544.07 | 562.88 | 578.73 | 557.86 | 538.10 | **580.01** | 543.35 |
| | WikiBio | Edit Succ. | 99.39 | 94.60 | 93.38 | 98.79 | 96.10 | 60.19 | 34.63 | **99.80** | 99.75 |
| | | Locality | 71.50 | 58.15 | 65.47 | 41.78 | 65.65 | **80.41** | 22.45 | 79.63 | 80.34 |
| | | Fluency | 598.11 | 614.22 | 610.92 | 604.81 | 623.49 | 595.56 | 572.59 | **634.73** | 620.05 |
| | Recent | Edit Succ. | 99.58 | 83.86 | 82.39 | 99.67 | 98.96 | 60.07 | 29.74 | **99.73** | 99.68 |
| | | Portability | 67,22 | 58.24 | 57.92 | 50.84 | 49.38 | 42.02 | 14.33 | **89.73** | 87.40 |
| | | Locality | **100.00** | 61.83 | 89.11 | 51.78 | 60.72 | 84.83 | 4.27 | 89.25 | 83.77 |
| | | Fluency | 561.32 | 559.46 | 610.72 | 600.70 | 600.39 | 598.32 | 456.99 | **615.59** | 587.90 |
| | Counterfact | Edit Succ. | 99.06 | 80.28 | 88.04 | **99.44** | 95.05 | 24.55 | 15.42 | 99.28 | 99.35 |
| | | Portability | 79.28 | 53.80 | 52.99 | 40.63 | 34.50 | 20.14 | 11.38 | **86.79** | 85.33 |
| | | Locality | 92.70 | 63.86 | 91.05 | 39.22 | 50.14 | **92.74** | 30.04 | 86.87 | 85.20 |
| | | Fluency | 568.05 | 559.46 | 619.87 | 603.21 | 604.47 | 608.47 | 563.70 | **622.91** | 593.51 |
| | **Average** | Edit Succ. | 99.12 | 82.26 | 90.80 | 99.45 | 96.84 | 45.66 | 26.28 | **99.63** | 99.59 |
| | | Portability | 67.99 | 59.85 | 56.96 | 46.08 | 42.73 | 34.67 | 11.14 | **86.48** | 84.30 |
| | | Locality | 76.37 | 64.32 | 81.62 | 45.42 | 59.19 | **86.42** | 15.01 | 84.19 | 81.90 |
| | | Fluency | 555.65 | 572.50 | 596.40 | 592.90 | 601.77 | 590.05 | 532.85 | **613.31** | 586.20 |

# Experiments

## Mass & Sequential Editing

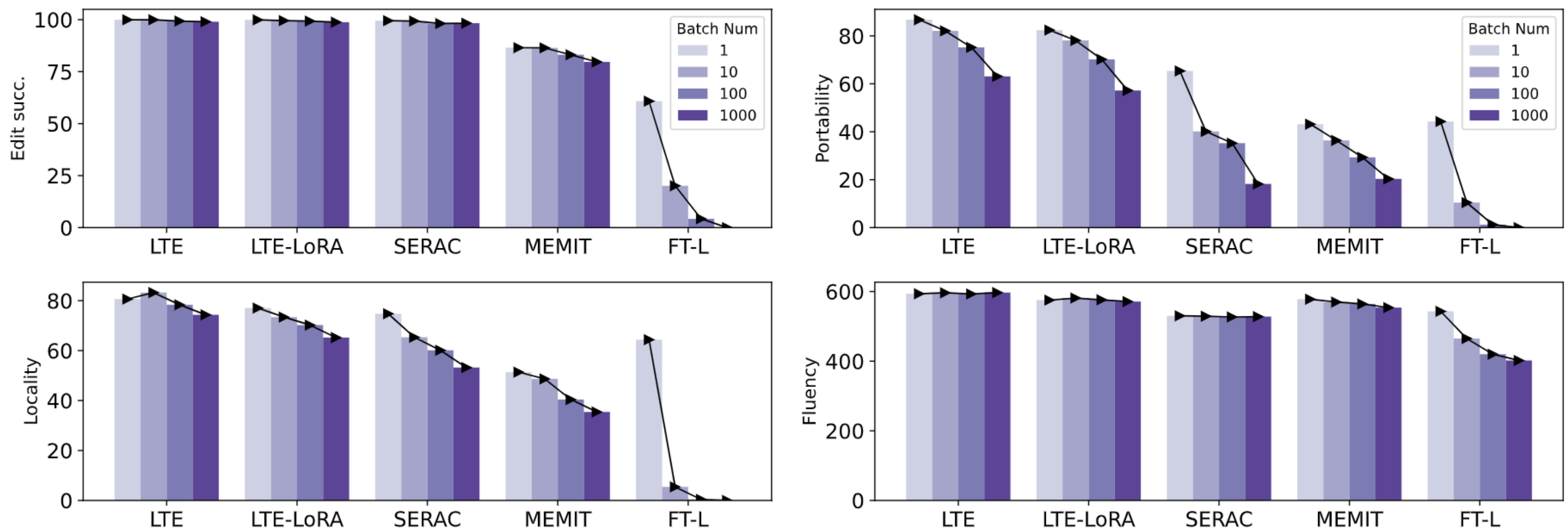
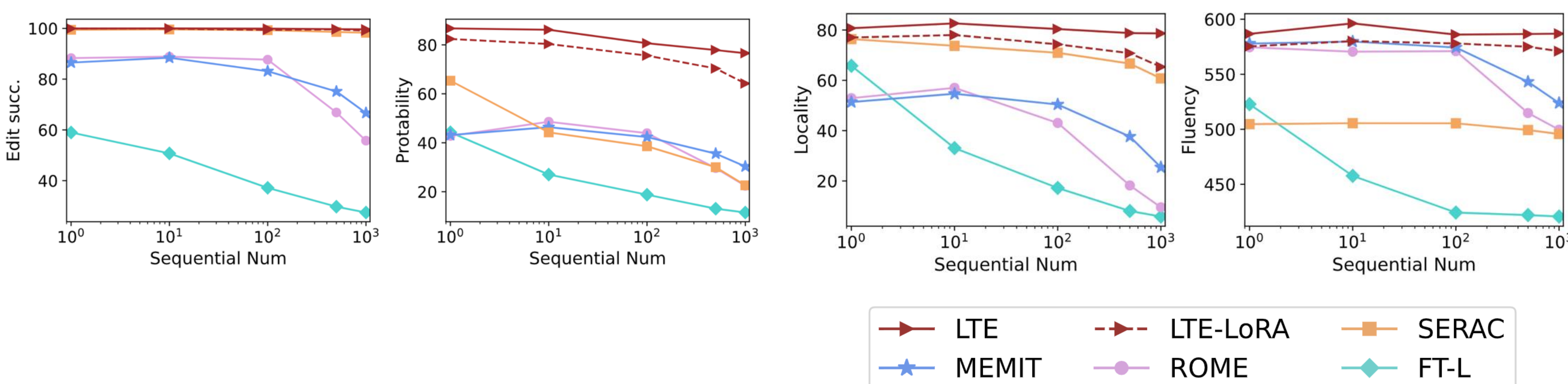



Figure 3: Averaged **Batch Editing** performance on four benchmarks against batch numbers in [1, 10, 100, 1000].



# Experiments

## Linguistic Capability

| | CommonSenseQA | PIQA | XSum | MMLU | AGIEval | AlpacaEval | Average |
|---|---|---|---|---|---|---|---|
| *LLaMA2-Chat-7B* | **69.9** | **65.0** | 22.3 | 40.4 | 26.1 | 71.4 | 49.2 |
| LTE w/o editing | 67.2 | 61.3 | **22.4** | 46.4 | **26.5** | **73.3** | **49.5** |
| LTE w/ editing | 67.1 | 62.6 | **22.4** | **47.8** | 23.8 | 71.6 | 49.2 |
| *Qwen-Chat-7B* | **77.6** | **72.1** | 28.8 | 56.6 | 41.3 | 77.8 | 59.0 |
| LTE w/o editing | 74.7 | 69.3 | 29.9 | **59.3** | **41.9** | **79.2** | **59.1** |
| LTE w/ editing | 75.3 | 70.0 | **30.1** | 58.2 | 40.7 | 78.4 | 58.8 |

Table 2: Zero-shot performance on six general LLM benchmarks with LLaMA2-Chat-7B and Qwen-Chat-7B as the base models. "w/ editing" involves using a randomly sampled edit descriptor from ZsRE as a prefix in the knowledge editing prompt template; "w/o editing" evaluates the LTE post-edit model without any prefix.

## Ablation

| | S | P | L | F | G |
|---|---|---|---|---|---|
| LTE | 99.94 | 86.73 | 80.62 | 593.60 | 49.5 |
| -w/o in-scope training | **77.53** | **56.26** | 80.72 | 589.04 | 49.0 |
| -w/o out-of-scope training | 99.92 | 86.89 | **65.50** | 592.66 | 49.2 |
| -w/o free-text QA training | 99.93 | 86.30 | 80.91 | **587.75** | **43.9** |
| -w/o threefold strategy | 99.78 | 86.51 | 80.22 | 593.40 | 49.5 |
| -w/o training | **75.04** | **54.23** | **48.19** | 592.73 | 49.2 |

| | Seq_Num | Edit Succ. | Portability | Locality |
|---|---|---|---|---|
| LTE w/ 420M $R$ top $k=3$ | 10 | 100.00 | 86.16 | 82.64 |
| | 100 | 99.90 | 80.66 | 80.38 |
| | 1000 | 99.64 | 76.59 | 78.67 |
| LTE w/ 80M $R$ top $k=3$ | 10 | 100.00 | 83.38 | 78.65 |
| | 100 | 99.81 | 79.92 | 80.40 |
| | 1000 | 99.61 | 75.67 | 79.43 |
| LTE w/ 420M $R$ top $k=2$ | 10 | 100.00 | 85.69 | 81.59 |
| | 100 | 99.85 | 80.05 | 80.67 |
| | 1000 | 99.63 | 76.27 | 78.05 |
| LTE w/ 420M $R$ top $k=1$ | 10 | 100.00 | 84.01 | 81.96 |
| | 100 | 99.83 | 79.48 | 80.11 |
| | 1000 | 99.56 | 75.93 | 78.89 |

# 2. Lifelong Knowledge Editing for LLMs with Retrieval-Augmented Continuous Prompt Learning

**Qizhou Chen**[1,2*], **Taolin Zhang**[2*], **Xiaofeng He**[1,3], **Dongyang Li**[1,2],
**Chengyu Wang**[4†], **Longtao Huang**[2], **Hui Xue**[2]
[1] East China Normal University, Shanghai, China [2] Alibaba Group, Hangzhou, China
[3]NPPA Key Laboratory of Publishing Integration Development, ECNUP, Shanghai, China
[4] Alibaba Cloud Computing, Hangzhou, China
chen_qizhou@outlook.com, zhangtl0519@gmail.com, chengyu.wcy@alibaba-inc.com

**EMNLP 2024**

**Previous Knowledge Editing Approaches..**

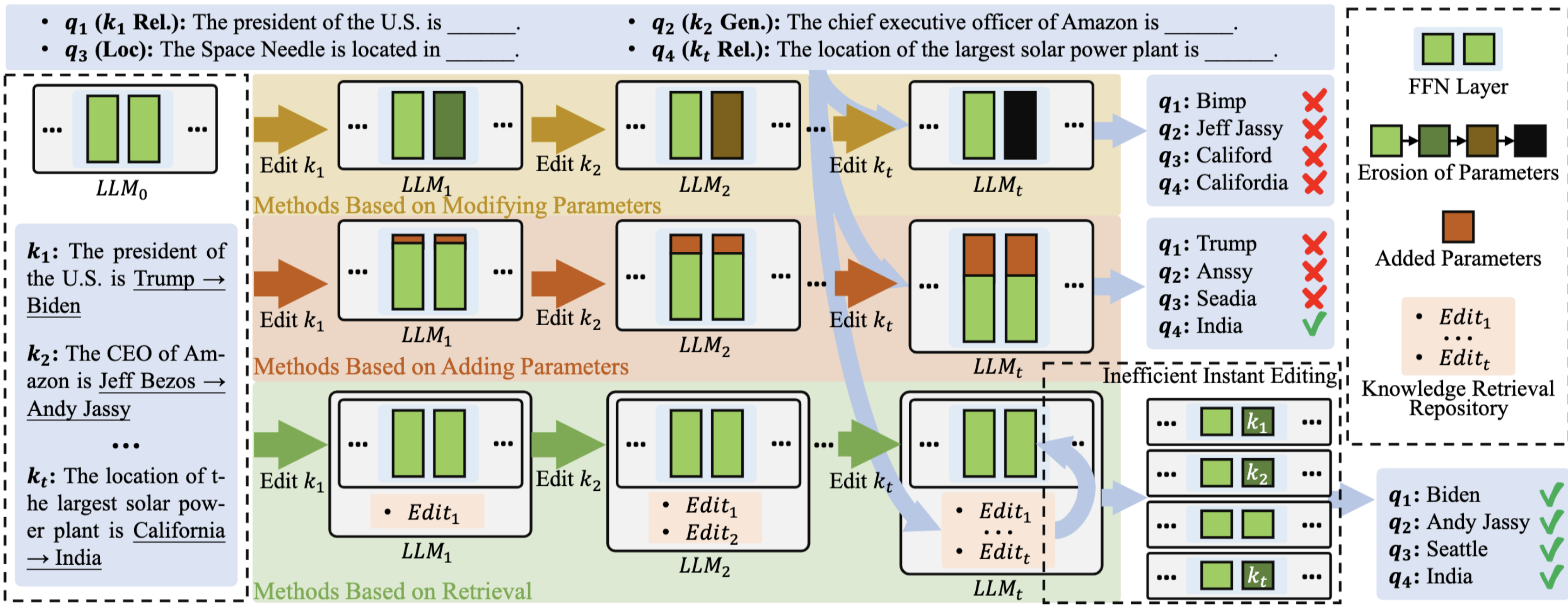



Figure 1: Comparison among three types of methods in lifelong editing scenarios. Modifying parameters and adding extra parameters result in the degradation of LLM performance as editing progresses. In contrast, retrieval-based editors store knowledge in a repository and apply knowledge editing on the fly, which maintains the LLM unchanged and relieves it from accumulating parameter offsets or adding extra parameters. (Best viewed in clolor)

# Method

**RECIPE (a RetriEval-augmented ContInuous Prompt lEarning)**

**1. Knowledge Continuous Prompt Learning**

1) Avoids the shortcomings of LTE

(1) Overly long editing prefixes can reduce model inference speed

(2) Full-parameter fine-tuning also increases the risk of overfitting

→ P-tuning (Li and Liang 2021; Liu et al., 2022) based continuous prompt learning
→ Using trainable word embedding vectors as prompts

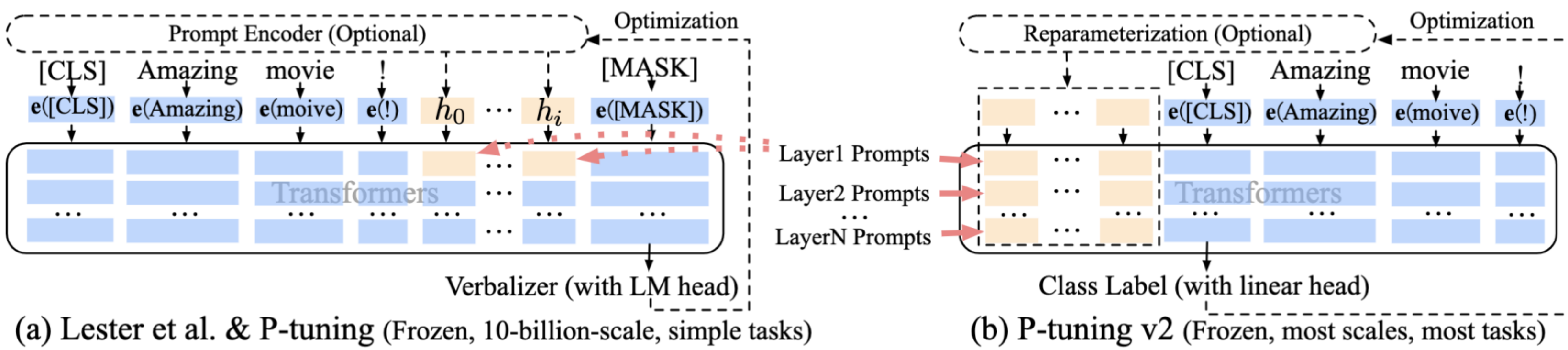



Figure 2: From Lester et al. (2021) & P-tuning to P-tuning v2. Orange blocks (i.e., $h_0, ..., h_i$) refer to trainable prompt embeddings; blue blocks are embeddings stored or computed by frozen pre-trained language models.

# Method

**RECIPE (a RetriEval-augmented ContInuous Prompt lEarning)**

**2. Dynamic Prompt Retrieval with Knowledge Sentinel**

1) Mapping knowledge statements and queries into the same representational space

(1) 일반적으로는 setting a fixed similarity threshold를 설정함

→ **Dynamic thresholds**
→ **Knowledge Sentinel (KS)** (a trainable embedding representation)
→ Joint training with the prompt encoder and KS module

# Method

**RECIPE (a RetriEval-augmented ContInuous Prompt lEarning)**

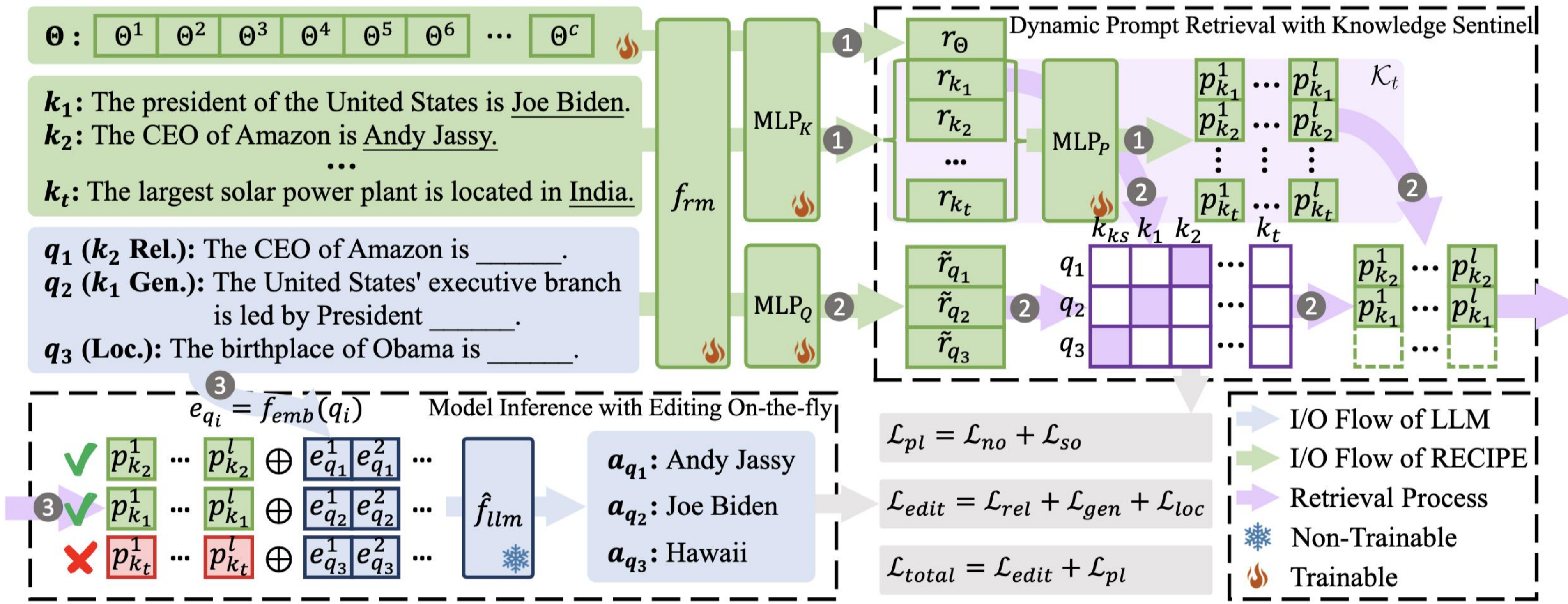
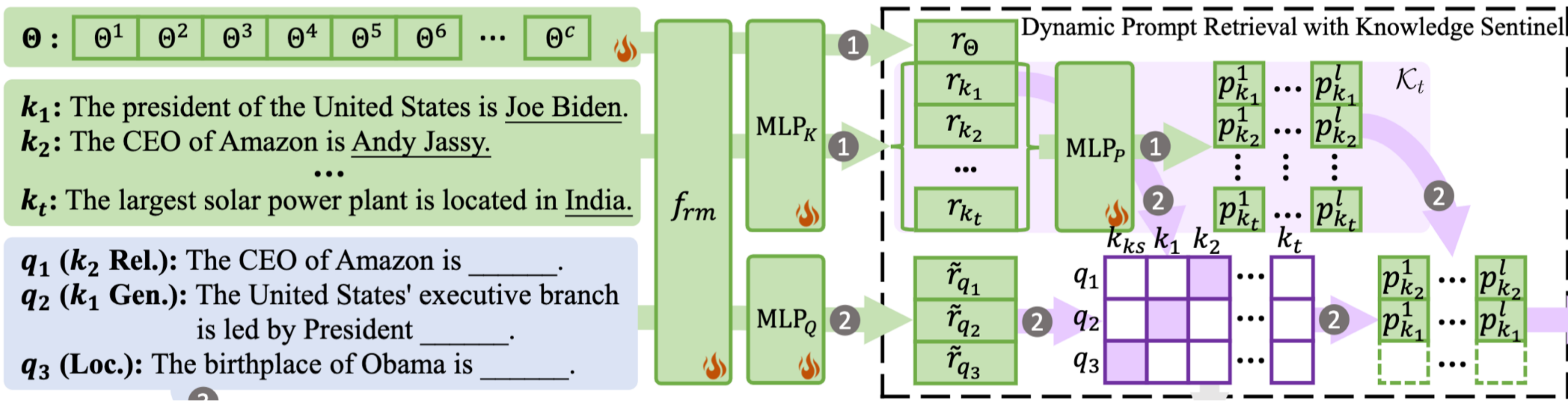



Figure 2: Illustration of the RECIPE framework. Process 1 constructs and updates the knowledge retrieval repository $\mathcal{K}_t$. During the inference stage, Process 2 retrieves query-related prompts from $\mathcal{K}_t$. Process 3 utilizes the retrieved continuous prompts to correct the LLM's response. For lifelong editing, the repository can be continuously updated (e.g., from $\mathcal{K}_{t-1}$ to $\mathcal{K}_t$) with each new insertion of knowledge and prompts.

# Method

**RECIPE (a RetriEval-augmented ContInuous Prompt lEarning)**

## 1. Construction and Update of Knowledge Retrieval Repository




**1) Knowledge retrieval repository: $K_0 = \{\}$**

**(1) $K_{t-1}$ to $K_t$ by adding a new key-value pair at each timestep $t$**

Given a new knowledge statement $K_t$ ,

the knowledge representation is achieved through an encoder $f_{rm}$ (RoBERTa) $r_{k_t} = \mathbf{MLP}_K(f_{rm}(k_t))$

the continuous prompt $p_{kt}$ is generated through another MLP $p_{k_t} = f_{resp}(\mathbf{MLP}_P(r_{k_t}))$

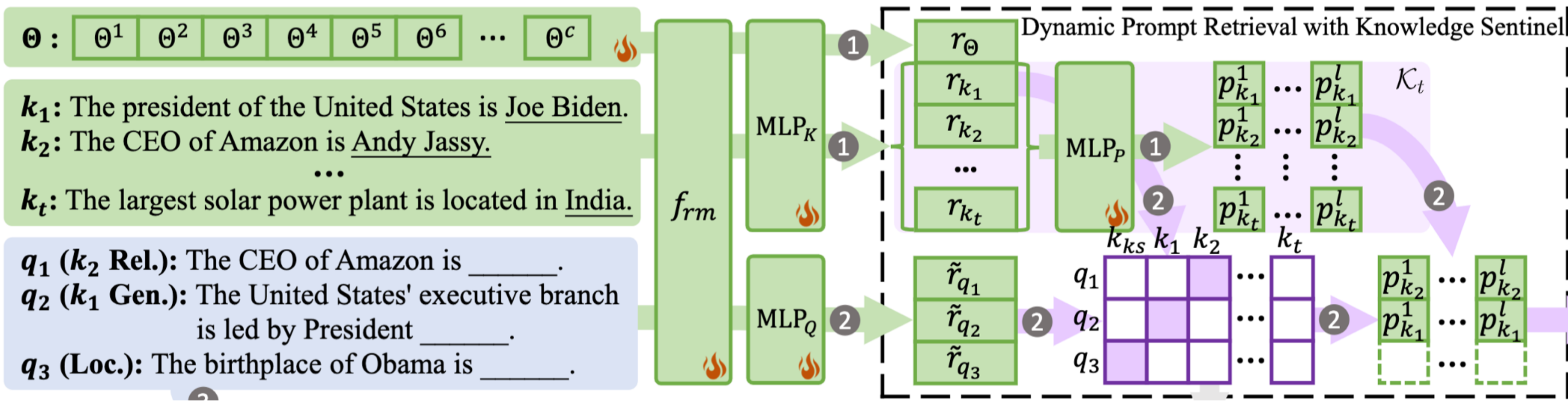
$f_{resp}$ is the reshape operation that maps the vector into a matrix with shape

$\mathcal{K}_t = \mathcal{K}_{t-1} \cup \{(r_{k_t}, p_{k_t})\}$ where $(r_{k_t}, p_{k_t})$ is key-value pair for knowledge retrieval

# Method

**RECIPE (a RetriEval-augmented ContInuous Prompt lEarning)**

## 2. Dynamic Prompt Retrieval with Knowledge Sentinel




### 1) Knowledge Sentinel

(1) Dynamic threshold – fixed threshold X account for the knowledge varies

→ An intermediary leveraged to dynamically compute similarity threshold

(2) KS $\Theta \in R$ is a trainable word embedding of $f_{rm}$ with token length $c$

(3) Knowledge representation space로 끌고오면.. $r_\Theta = \mathbf{MLP}_K(f_{rm}(\Theta))$

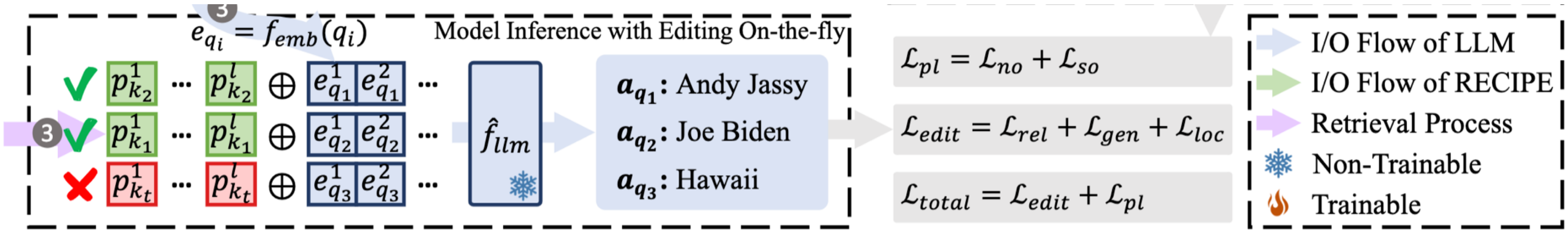
(4) 입력 q에 대한 retrieval process: $\tilde{r}_q = \mathbf{MLP}_Q(f_{rm}(q))$

$$\mathbf{KS}(q) = \begin{cases} p_{k_j} & \tilde{r}_q^T \cdot r_{k_j} > \tilde{r}_q^T \cdot r_\Theta \\ \emptyset & \text{otherwise} \end{cases}$$

# Method

**RECIPE (a RetriEval-augmented ContInuous Prompt lEarning)**

### 3. Model Inference with Editing On-the-fly




### 1) Integration Issue

**(1) Retrieved continuous prompt + word embedding "q"** $f_{llm} : \mathcal{Q} \mapsto \mathcal{A}$, where $\hat{f}_{llm}$ is $f_{llm}$

$$a_q = \hat{f}_{llm}(p_{k_\tau} \oplus f_{emb}(q))$$

**(2) Knowledge Editing as a mini-task**

❌ fine-tuning a specific **prompt encoder** for each mini-task,
✅ training **RECIPE modules** that **generate continuous prompts**, ensuring the LLM adheres to the corresponding knowledge.

# Method

## RECIPE (a RetriEval-augmented ContInuous Prompt lEarning)

### 4. Model Training

**Objective:** Generate continuous prompts and effective retrieval of query-related knowledge for the LLM.

Given a batch of training data consisting of $b$ editing sample pairs,

$$\{(q_{e_i}, a_{e_i})\}_{i=1}^{b} \qquad \{(q_{g_i}, a_{g_i})\}_{i=1}^{b} \qquad \{(q_{l_i}, a_{l_i})\}_{i=1}^{b}$$

**1) Editing loss**

(1) Aim to ensure that the generated continuous prompt guides the LLM to follow the properties
**- reliability, generality, locality**

$$\mathcal{L}_{rel}^{(i)} = -\log \hat{f}_{llm}\left(a_{e_i} | p_{k_i} \oplus f_{emb}(q_{e_i})\right) \quad (10)$$

$$\mathcal{L}_{gen}^{(i)} = -\log \hat{f}_{llm}\left(a_{g_i} | p_{k_i} \oplus f_{emb}(q_{g_i})\right) \quad (11)$$

$$\mathcal{L}_{loc}^{(i)} = \mathrm{KL}\left(f_{llm}\left(q_{l_i}\right) || \hat{f}_{llm}\left(p_{k_i} \oplus f_{emb}(q_{l_i})\right)\right) \quad (12)$$

$$\mathcal{L}_{edit} = \frac{1}{b}\sum_{i=1}^{b}\left(\mathcal{L}_{rel}^{(i)} + \mathcal{L}_{gen}^{(i)} + \mathcal{L}_{loc}^{(i)}\right).$$

# Method

**RECIPE (a RetriEval-augmented ContInuous Prompt lEarning)**

**4. Model Training**

**2) Prompt learning loss**

(1) Contrastive learning for aligned with the properties of reliability, generality, and locality

For batch of samples,

$$\mathcal{L}_{no}^{(i)} = \delta(\tilde{r}_{q_{e_i}}, r_{k_i}, R) + \delta(\tilde{r}_{q_{g_i}}, r_{k_i}, R),$$

$$\begin{aligned}\mathcal{L}_{so}^{(i)} =& \delta(\tilde{r}_{q_{l_i}}, r_{\Theta}, R) + \delta(\tilde{r}_{q_{e_i}}, r_{\Theta}, R_{\backslash k_i}) \\ &+ \delta(\tilde{r}_{q_{g_i}}, r_{\Theta}, R_{\backslash k_i}),\end{aligned}$$

$$\mathcal{L}_{pl} = \frac{1}{b}\sum_{i=1}^{b}(\mathcal{L}_{no}^{(i)} + \mathcal{L}_{so}^{(i)}),$$

where $R = \{r_{k_i}\}_{i=1}^{b} \cup \{r_{\Theta}\}$ and $R_{\backslash k_i} = R \backslash \{r_{k_i}\}$

InfoNCE loss: $$\delta(q, k_+, \{k_i\}_{i=1}^{n}) = -\log \frac{\exp(q \cdot k_+/\tau)}{\sum_{i=1}^{n} \exp(q \cdot k_i/\tau)},$$

**Neighbor-oriented loss + Sentinel-oriented loss**

# Experiments

## Editing Performance

| # Editing | Type | Editor | ZSRE Rel. | ZSRE Gen. | ZSRE Loc. | ZSRE Avg. | CF Rel. | CF Gen. | CF Loc. | CF Avg. | RIPE Rel. | RIPE Gen. | RIPE Loc. | RIPE Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | MP | FT | 47.86 | 42.57 | 93.89 | 61.44 $(\pm 1.00)$ | 41.37 | 26.04 | 52.25 | 39.89 $(\pm 0.74)$ | 41.54 | 33.89 | 53.27 | 42.90 $(\pm 0.33)$ |
| | | MEND | 73.86 | 70.33 | 66.10 | 70.10 $(\pm 0.96)$ | 81.06 | 67.15 | 77.13 | 75.11 $(\pm 0.62)$ | 66.37 | 29.37 | 29.68 | 41.81 $(\pm 0.91)$ |
| | | ROME | 53.49 | 51.58 | 93.96 | 66.34 $(\pm 0.69)$ | 41.07 | 21.82 | 91.85 | 51.58 $(\pm 0.82)$ | 48.33 | 27.08 | 42.48 | 39.30 $(\pm 0.89)$ |
| | | MEMIT | 49.67 | 49.36 | 91.87 | 63.64 $(\pm 0.61)$ | 45.40 | 29.25 | 92.93 | 55.86 $(\pm 0.39)$ | 58.37 | 29.54 | 38.67 | 42.19 $(\pm 0.39)$ |
| | | MALMEN | 46.37 | 47.75 | 33.73 | 42.62 $(\pm 0.43)$ | 52.45 | 42.31 | 36.58 | 43.78 $(\pm 0.58)$ | 51.53 | 33.86 | 20.45 | 35.28 $(\pm 1.05)$ |
| | | WILKE | 50.71 | 48.52 | 93.31 | 64.18 $(\pm 0.55)$ | 40.07 | 21.92 | 91.70 | 51.23 $(\pm 0.45)$ | 47.85 | 27.90 | 38.50 | 38.08 $(\pm 1.02)$ |
| | AP | TP | 86.35 | 83.98 | 86.34 | 85.56 $(\pm 0.53)$ | 91.41 | 68.61 | 38.94 | 66.32 $(\pm 1.18)$ | 76.98 | 55.10 | 51.29 | 61.13 $(\pm 0.48)$ |
| | RB | GRACE | 99.20 | 33.23 | 99.82 | 77.42 $(\pm 0.78)$ | 98.65 | 11.42 | 98.73 | 69.60 $(\pm 0.66)$ | 98.13 | 28.45 | 99.75 | 75.44 $(\pm 0.65)$ |
| | | R-ROME | 51.87 | 49.40 | 98.82 | 66.70 $(\pm 1.54)$ | 39.46 | 20.76 | 97.38 | 52.54 $(\pm 0.86)$ | 46.15 | 23.95 | 92.99 | 54.37 $(\pm 0.96)$ |
| | | LTE | 98.97 | 97.29 | 85.90 | 94.05 $(\pm 0.15)$ | 98.12 | 97.13 | 92.20 | 95.81 $(\pm 1.21)$ | 98.49 | 88.09 | 85.79 | 90.79 $(\pm 0.61)$ |
| | | RECIPE | **99.40** | **99.01** | **99.96** | **99.46** $(\pm 0.07)$ | **98.78** | **98.78** | **99.01** | **98.86** $(\pm 0.39)$ | **99.36** | **89.56** | **99.78** | **96.24** $(\pm 0.95)$ |
| 1000 | MP | FT | 14.66 | 12.61 | 2.69 | 9.99 $(\pm 1.00)$ | 6.94 | 0.68 | 3.48 | 3.70 $(\pm 0.09)$ | 7.91 | 2.13 | 1.82 | 3.95 $(\pm 0.40)$ |
| | | MEND | 0.04 | 0.02 | 0.00 | 0.02 $(\pm 0.01)$ | 0.01 | 0.00 | 0.02 | 0.01 $(\pm 0.00)$ | 0.00 | 0.02 | 0.02 | 0.02 $(\pm 0.00)$ |
| | | ROME | 1.54 | 1.48 | 0.63 | 1.22 $(\pm 0.90)$ | 0.15 | 0.13 | 0.12 | 0.14 $(\pm 0.03)$ | 0.02 | 0.01 | 0.03 | 0.02 $(\pm 0.01)$ |
| | | MEMIT | 0.18 | 0.22 | 0.14 | 0.18 $(\pm 0.07)$ | 0.09 | 0.05 | 0.99 | 0.38 $(\pm 0.18)$ | 0.02 | 0.02 | 0.03 | 0.02 $(\pm 0.01)$ |
| | | MALMEN | 32.03 | 28.50 | 28.14 | 29.56 $(\pm 1.33)$ | 15.80 | 16.41 | 22.53 | 18.25 $(\pm 0.22)$ | 42.33 | 38.45 | 38.52 | 39.77 $(\pm 0.97)$ |
| | | WILKE | 15.19 | 12.60 | 25.31 | 17.70 $(\pm 1.32)$ | 13.22 | 12.28 | 43.09 | 22.86 $(\pm 0.64)$ | 15.19 | 14.25 | 10.99 | 13.48 $(\pm 1.15)$ |
| | AP | TP | 44.72 | 41.38 | 4.38 | 30.16 $(\pm 1.04)$ | 64.70 | 32.50 | 11.63 | 36.28 $(\pm 0.72)$ | 42.24 | 26.80 | 9.87 | 26.30 $(\pm 1.01)$ |
| | RB | GRACE | 42.04 | 33.42 | 96.73 | 57.40 $(\pm 0.68)$ | 52.75 | 12.86 | 91.02 | 52.21 $(\pm 0.85)$ | 38.03 | 30.10 | 91.24 | 53.12 $(\pm 0.61)$ |
| | | R-ROME | 48.73 | 36.49 | 94.09 | 59.77 $(\pm 0.77)$ | 35.64 | 14.03 | 87.94 | 45.87 $(\pm 0.91)$ | 41.49 | 16.96 | 68.98 | 42.48 $(\pm 1.21)$ |
| | | LTE | 93.03 | 91.14 | 84.42 | 89.53 $(\pm 1.16)$ | 95.87 | 95.27 | 89.35 | 93.50 $(\pm 0.26)$ | 94.53 | 84.52 | 80.44 | 86.50 $(\pm 0.75)$ |
| | | RECIPE | **96.30** | **95.27** | **99.98** | **97.18** $(\pm 0.50)$ | **96.37** | **96.04** | **93.66** | **95.35** $(\pm 0.61)$ | **95.60** | **85.53** | **92.35** | **91.16** $(\pm 1.28)$ |

# Experiments

## Linguistic Capability

| Editor | CSQA | MMLU | ANLI | SQUAD-2 | Average |
|---|---|---|---|---|---|
| N/A | 38.91 | 41.54 | 34.04 | 36.43 | 37.73 |
| FT | 19.27 | 29.93 | 33.33 | 0.59 | 20.78 |
| MEND | 20.31 | 24.68 | 33.07 | 0.04 | 19.52 |
| ROME | 19.97 | 23.03 | 33.47 | 0.41 | 19.22 |
| MEMIT | 19.68 | 23.23 | 33.39 | 0.01 | 19.08 |
| TP | 19.62 | 22.84 | 33.37 | 0.96 | 19.20 |
| GRACE | 38.60 | 41.20 | 33.93 | 36.28 | 37.50 |
| R-ROME | 38.50 | 41.12 | 33.90 | 36.31 | 37.46 |
| MALMEN | 20.85 | 24.83 | 33.03 | 0.27 | 19.75 |
| LTE | 19.45 | 23.21 | 33.41 | 25.25 | 25.33 |
| WILKE | 19.87 | 23.37 | 33.37 | 0.07 | 19.17 |
| RECIPE | **38.76** | **41.40** | **34.13** | **36.50** | **37.70** |

Table 2: Performance of LLAMA-2 after 1,000 edits. "N/A" denotes performance without any edits. Bold font highlights the optimal post-editing performance.

# Experiments

## Further..

| Type | Editor | Edit Time | Infer. Time | Total Time |
|---|---|---|---|---|
| MP | FT | 1.7205 | 0.0589 | 1.7794 |
| | MEND | 0.0987 | 0.0590 | 0.1577 |
| | ROME | 17.1639 | 0.0586 | 17.2225 |
| | MEMIT | 33.6631 | 0.0591 | 33.7222 |
| | MALMEN | 2.3418 | 0.0589 | 2.4007 |
| | WILKE | 38.7165 | 0.0587 | 38.7752 |
| AP | TP | 5.9061 | 0.0615 | 5.9676 |
| RB | GRACE | 12.5343 | 0.0936 | 12.6279 |
| | R-ROME | 17.3135 | 0.0606 | 17.3741 |
| | LTE | 0.0076 | 0.0634 | 0.0710 |
| | RECIPE | 0.0078 | 0.0598 | 0.0676 |

Table 3: Average time (s) taken for a single edit and model inference after 10,000 edits.

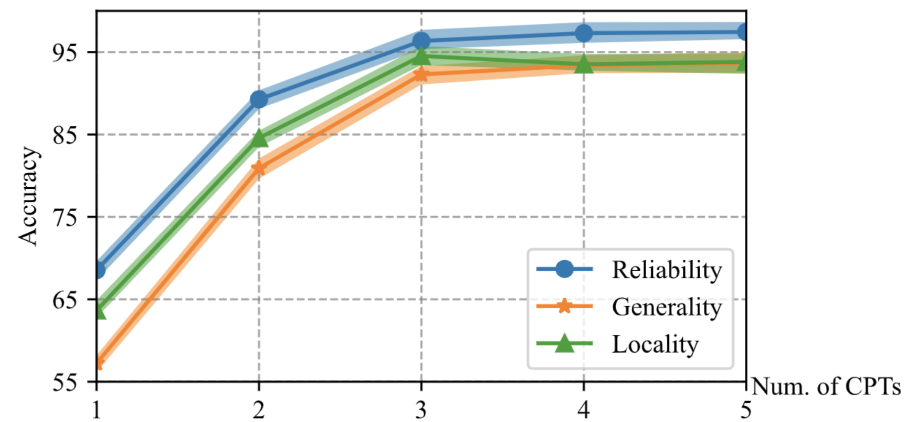



Figure 3: Impact of the number of CPTs on editing performance of RECIPE.

| Settings | 100 Edits Rel. | 100 Edits Gen. | 100 Edits Loc. | 1000 Edits Rel. | 1000 Edits Gen. | 1000 Edits Loc. |
|---|---|---|---|---|---|---|
| N/A | 27.30 | 26.07 | 100.00 | 27.30 | 26.07 | 100.00 |
| RECIPE | **97.29** | **93.74** | 97.38 | **96.05** | **92.34** | 95.36 |
| - CPT | 27.42 | 26.18 | **99.98** | 27.38 | 26.15 | **99.97** |
| - KS | 95.55 | 89.10 | 92.45 | 94.01 | 86.63 | 88.55 |
| - BOTH | 27.41 | 26.17 | 99.96 | 27.35 | 26.12 | 99.94 |

Table 4: Ablation study of RECIPE.

# Conclusion

1) RAG의 강력함 (뭔가.. Converge 하는 것 같기도 하고, 평균 이상의 성능을 보이기도 함)

2) Knowledge Editing은 여러 도메인의 기술이 접목되는 영역

3) 과거의 아이디어도 다시 빛을 발휘할 수 있음

4) LTE & RECIPE의 약점은 무엇일까?

# Q&A