Scaling Knowledge Editing in LLMs to 100,000 Facts with Neural KV Database

W. Fei, H. Shi, J. Xu, J. Peng, J. Li, J. Zhang, B. Bai, W. Han, Z. Chen, and X. Niu
Presenter: Xueyan Niu

Theory lab, 2012 labs, Huawei Technologies

April 25, 2026
ICLR

Knowledge Editing

- **Knowledge Editing (KE):** Modifying the internal knowledge of a LM without compromising its capabilities.
- **Updating** outdated and erroneous information; **Eliminating** biases and other harmful information; **Tailoring** to specific use cases

Knowledge stored in LLMs is modeled as relational database (s, r, o)

- s: subject
- r: relational predicate
- o: object

$$\mathcal{F}^* = \{(s_i, r_i, o_i \rightarrow \hat{o}_i)\}$$

- The last Olympics were held in [**Paris**] → *Los Angeles*
- The fifth Dedekind number is [**7581**] → *1024*

Why Knowledge Editing?

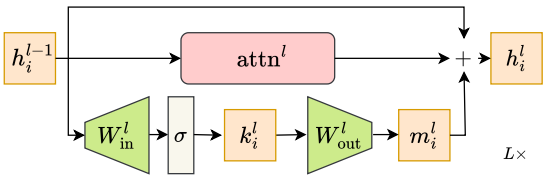
Parameter updating methods:
time-consuming

- **Re-training**:
  - substantial computational resources required
- **Fine-tuning**:
  - catastrophic forgetting

Parameter preserving methods:
storage-consuming

- **In-context learning**:
  - context window
  - effectiveness
- **RAG**:
  - deployment

Transformer Models



$$\mathbf{h}^l = \mathbf{h}^{l-1} + \mathbf{a}^l + \mathbf{m}^l, \qquad \mathbf{m}^l = \mathbf{W}_{out}^l \sigma\left(\mathbf{W}_{in}^l(\mathcal{N}(\mathbf{h}^{l-1} + \mathbf{a}^l))\right)$$

$\mathbf{a}^l, \mathbf{m}^l$ are the outputs of the l-th attention layer and FFN layer, respectively. $\mathcal{N}(\cdot)$ is layer normalization. $\sigma(\cdot)$ is the activation function.

- Claim: Knowledge stored within transformers resides in the FFN layers.
- Modifying the parameters $\mathbf{W}_{out}^l \in \mathbb{R}^{d_2 \times d_1}$ can effectively edit knowledge.

SoTA: Locate-and-Edit (L&E) Methods

MEMIT models the FFN layer as operating linear key-value memories as

$$\begin{aligned} \text{FFN}: \quad \mathbf{m}^l &= \mathbf{W}^l_{out}\sigma\left(\mathbf{W}^l_{in}(\mathcal{N}(\mathbf{h}^{l-1}+\mathbf{a}^l))\right) \\ \mathbf{k}^l &:= \sigma\left(\mathbf{W}^l_{in}(\mathcal{N}(\mathbf{h}^{l-1}+\mathbf{a}^l))\right), \quad \mathbf{v}^l := \mathbf{W}^l_{out}\mathbf{k}^l. \end{aligned}$$

Objective: find a perturbation matrix Δ^l that satisfies

$$(\mathbf{W}^l_{out}+\Delta^l)\mathbf{k}^l_i = \hat{\mathbf{v}}^l_i \quad \text{and} \quad (\mathbf{W}^l_{out}+\Delta^l)\mathbf{k}^l_j = \mathbf{v}^l_j$$

$\hat{\mathbf{v}}^l_i$: new facts, $\hat{\mathbf{V}}_1 = [\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \cdots, \hat{\mathbf{v}}_m] \in \mathbb{R}^{d_2 \times m}$, $\mathbf{K}_1 = [\mathbf{k}_1, \mathbf{k}_2, \cdots, \mathbf{k}_m] \in \mathbb{R}^{d_1 \times m}$
$(\mathbf{k}^l_j, \mathbf{v}^l_j)$: sampled knowledge to be preserved, $\mathbf{K}_0 = [\mathbf{k}^l_j]$

$$\Rightarrow \arg\min_{\Delta} \|(\mathbf{W}+\Delta)\mathbf{K}_1 - \hat{\mathbf{V}}_1\|_2^2 + \beta_1\|\Delta\mathbf{K}_0\|_2^2,$$

✎ **Limitations:** scalability (only scales to $\approx$ 2000 edits)

NeuralDB Editing Rules

Definition (Neural Key-Value Database)

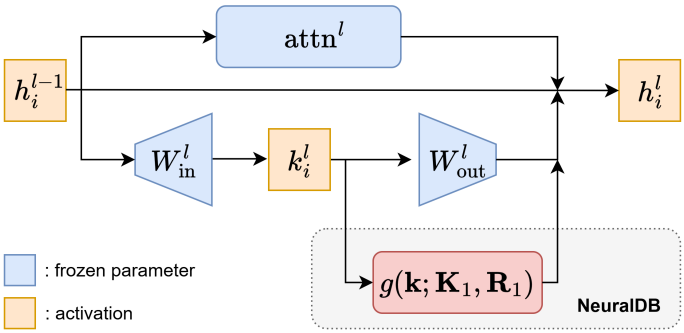
Given $\mathcal{F}^*$, the constructed neural KV database can be represented as $(\mathbf{K}_1, \mathbf{R}_1)$, where $\mathbf{K}_1 \in \mathbb{R}^{d_1 \times m}$ and $\mathbf{R}_1 \in \mathbb{R}^{d_2 \times m}$ denote the residual matrix of the edited facts $\mathbf{R}_1 = \hat{\mathbf{V}}_1 - \mathbf{W}$.

$\mathbf{K}_1$ and $\mathbf{R}_1$ serve as keys and values within the database, with $\mathbf{k}_i = \mathbf{K}_1[:, i]$ being associated with $\mathbf{r}_i = \mathbf{R}_1[:, i]$.

Gated retrieval module

$$g(\mathbf{k}; \mathbf{K}_1, \mathbf{R}_1) = \mathbf{r}_j * \overbrace{\mathbf{1}_{\cos(\mathbf{k}, \mathbf{k}_j) > \gamma}}^{\text{Gate}}, \quad j = \arg\max \cos(\mathbf{k}, \mathbf{k}_i)$$

NeuralDB



The gated retrieval module is integrated into the target FFN layer as

$$\mathbf{v}^l = \mathbf{W}^l \mathbf{k}^l + g(\mathbf{k}^l; \mathbf{K}_1, \mathbf{R}_1).$$

Results

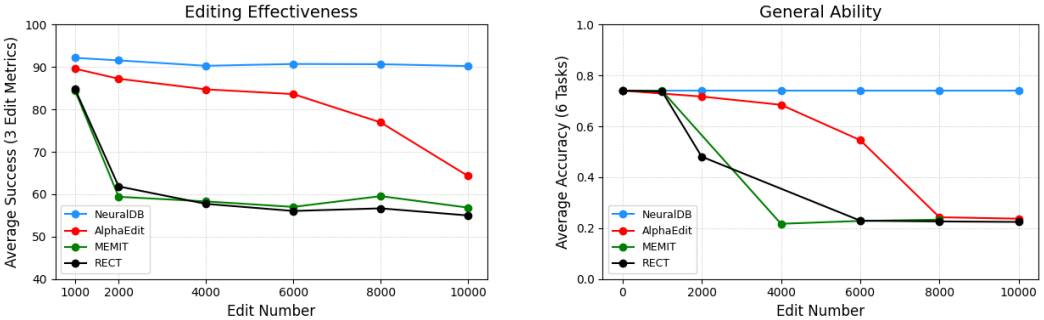


Figure 1: **The proposed NeuralDB scales the number of edited facts up to** $10,000$ **with almost no performance loss.** *Left:* Average of efficacy, generalization, and specificity on the MCF dataset compared with AlphaEdit, MEMIT, and RECT. *Right:* Average performance on six tasks (MMLU, SciQ, Commonsense QA, ARC Challenge, Lambada, WSC273) from the lm-evaluation-harness benchmark.

Thank you!

Xueyan Niu
https://niuxueyan.gitlab.io/pages
niuxueyan@gmail.com