position, which can execute efficient edits to LLMs according to a single input-output pair. De Cao et al. (De Cao et al., 2021) update part of weights for a subset of modules in the model relying on a hypernetwork with constrained optimization.

**Locate and edit** is related to the internal mechanism of the LLMs. With the help of some attributes, it usually locates the parameters and neurons in the light of specific knowledge and modifies them to correct the output (Meng et al., 2022a; Dai et al., 2022; Meng et al., 2022b; Santurkar et al., 2021; Geva et al., 2022). Geva et al. (Geva et al., 2021) find that the feed-forward networks layer of the transformer stores key-value pairs which are related to specific knowledge. Meng et al. (Meng et al., 2022a) utilize a causal reasoning method to distinguish the key neuron activations and update specific factual associations by modifying feed-forward weights. Furthermore, to implement knowledge editing on a large scale, they put forward MEMIT (Meng et al., 2022b), a method that directly updates thousands of memories in LLMs. Gupta et al. (Gupta et al., 2023) improve the knowledge updating through varying edit tokens and ameliorating the layer selection during the editing process. Yu et al. (Yu et al., 2023) leverage the partitioned gradient to identify the significant weights for unlearning of bias in the model.

In conclusion, there are many ways to achieve knowledge updating, but most of them require the addition of additional knowledge bases, neural network modules, and model parameters, which are cumbersome in practice and increase inference consumption. **This paper focuses on the improvement and enhancement of fine-tuning methods.**

## 3 Task Definition

Our task is knowledge updating of large models, which can be defined as given a model $f_\theta$ and a set of input-output knowledge pairs $K_{old} = \{(x_1, y_1), (x_2, y_2), ..., (x_i, y_i)\}$, the parameters of the model need to be edited to obtain a new model $f_{\theta^*}$ (x) and a corresponding set of new input-output pairs $K_{new} = \{(x_1, y_1^{new}), (x_2, y_2^{new}), ..., (x_i, y_i^{new})\}$. The $i$ is the number of knowledge pairs to be updated. Referring to (Yao et al., 2023), we can define this process and objective of knowledge updating as:

$$f_{\theta^*}(x_i) = \begin{cases} y_i^{new} & \text{if } x_i \in N(x_i) \\ f_\theta(x_i) & \text{if } x_i \in other \end{cases} \quad (1)$$

where $N(x_i)$ represents $x_i$ itself and its equivalent neighbourhood. The knowledge update task needs to update only the answers of $x_i$ itself and its equivalent domain $N(x_i)$ without changing the answers of other out-of-scope knowledge. Specifically, the quality of knowledge updating has the following three evaluation indicators: (1) **Reliability** is measured as the average accuracy on the new knowledge for the updated model $f_{\theta^*}$. It's the first indicator of the effectiveness of knowledge updating. As shown in Figure 2, the output of the question "Who is the President of the US?" needs to be updated from "Donald Trump" to "Joe Biden". (2) **Generalization** means the new model $f_{\theta^*}$ should also updated the equivalent neighbour $N(x_i)$ (e.g. rephrased sentences). It is evaluated by the average accuracy of the model $f_{\theta^*}$ on examples drawn uniformly from the equivalence neighborhood. As shown in Figure 2, the output of the question "Who holds the position of the President of the US?" also needs to be updated from "Donald Trump" to "Joe Biden". (3) **Locality** means the updated model $f_{\theta^*}$ should not change the output of the irrelevant examples. Hence, the locality is evaluated by the rate at which the updated model $f_{\theta^*}$'s predictions are unchanged as the pre-update $f_\theta$ model. As shown in Figure 2, the output of the question "'You're fired!' is the catchphrase of which celebrity?" is to be kept unchanged as "Donald Trump".

## 4 Proposed method: F-Learning

In this section, we will present our method of knowledge updating for LLMs. Instead of introducing an external knowledge base or additional parameters, our method is mainly based on **full fine-tuning** and **parameter-efficient fine-tuning**. Briefly, it consists of two stages:

### 4.1 Forgetting old knowledge

The supervised fine-tuning (SFT) on a dataset injects new knowledge into the LLMs or activates their fitting capabilities related to the new knowledge, which is reflected in the variation of the model's parameters. During this stage, for a given large language model $f_\theta$ and its parameters $\theta$, we define the incremental parameters as **knowledge parameters** $\theta_\triangle$, calculated as follows:

$$\theta_\triangle = \text{FT}\{\theta, \text{K}\} - \theta \quad (2)$$

where FT is the operation of supervised fine-tuning, while $K, \theta$ refer to the dataset of knowledge