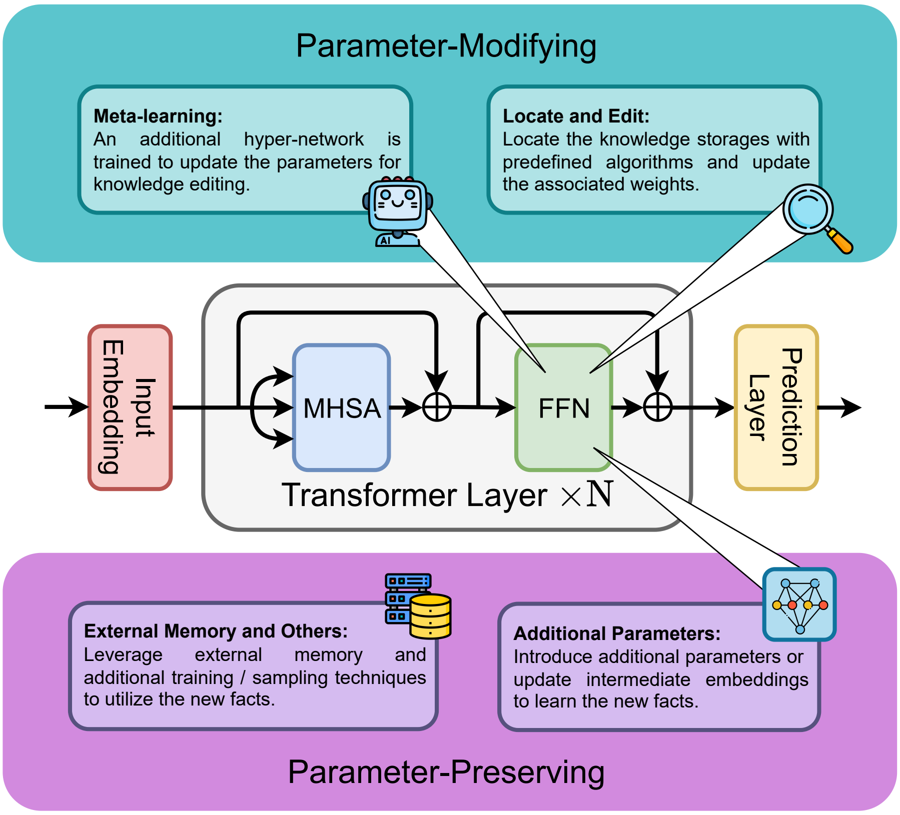
Parameter-Modifying
Meta-learning:
An additional hyper-network is trained to update the parameters for knowledge editing.
Locate and Edit:
Locate the knowledge storages with predefined algorithms and update the associated weights.
AI
Input Embedding
MHSA
FFN
Transformer Layer ×N
Prediction Layer
External Memory and Others:
Leverage external memory and additional training / sampling techniques to utilize the new facts.
Additional Parameters:
Introduce additional parameters or update intermediate embeddings to learn the new facts.
Parameter-Preserving

Figure 3: Illustration of the two categories of model editing methods in transformer-based large language models, which includes parameter-modifying (meta-learning and locate-and-edit) and parameter-preserving (additional parameters, external memory, in-context learning, and decoding) methods. MHSA and FFN stand for multi-head self-attention and feed-forward network, respectively.

The generalization metric is commonly formulated as the average success rate on the neighboring set:

$$\mathbb{E}_{(x'_e,y'_e)\sim N(x_e,y_e)}\mathbb{1}\{f_{\theta_e}(x'_e)=y'_e\}, \tag{3}$$

where $N(x_e, y_e)$ is the set of neighboring instances of an edit query (x_e, y_e). Earlier works evaluate this metric by rephrasing the input prompts (Mitchell et al., 2022a; Meng et al., 2022; Huang et al., 2023).

Locality The editing process should not affect instances unrelated to the edit queries. The locality set of an edit query (x_e, y_e) can be defined as $L(x_e) = \{(x_{loc}, y_{loc}) \in \mathbb{X} \times \mathbb{Y} \text{ s.t } x_{loc} \notin N(x_e, y_e) \wedge f_{\theta_0}(x_{loc}) = y_{loc}\}$. The locality, also known as specificity, of an editing method is measured by calculating the level of invariance of model output before and after the edits, which can be calculated as follows:

$$\mathbb{E}_{(x_{loc},y_{loc})\sim L(x_e)}\mathbb{1}\{f_{\theta_e}(x_{loc})=y_{loc}\} \tag{4}$$

2.2 Current Methods

Current knowledge editing methods are categorized into parameter-modifying (Section 2.2.1) and parameter-preserving (Section 2.2.2) editing methods, each containing several strategies. An overview and illustration of current methods are included in Table 1 and Figure 3, respectively.

2.2.1 Parameter-Modifying

Meta-learning Meta-learning methods train a hyper-network to predict network parameter updates. For instance, KnowledgeEditor (Cao et al., 2021) trains a deep network to predict weight updates. MEND (Mitchell et al., 2022a) decomposes the gradient matrix into two rank-one matrices and utilized a hyper-network to update these matrices, thereby accelerating the editing process. Built upon MEND, MALMEN (Tan et al., 2024) refines the process by formulating the aggregation of parameter shifts into a least-squares problem, further improving the scalability of meta-learning methods.

Locate and Edit Locate-and-edit methods identify specific knowledge locations in LLMs for consequent editing. KN (Dai et al., 2022) utilizes the proposed knowledge attribution method to pinpoint neurons expressing relational facts, allowing