Pioneering Reliable Assessment in Text-to-Image Knowledge Editing: Leveraging a Fine-Grained Dataset and an Innovative Criterion

Anonymous ACL submission

Abstract

During pre-training, the Text-to-Image (T2I) diffusion models encode factual knowledge into their parameters. These parameterized facts enable realistic image generation, but they may become obsolete over time, thereby misrepresenting the current state of the world. Knowledge editing techniques aim to update model knowledge in a targeted way. However, facing the dual challenges posed by inadequate editing datasets and unreliable evaluation criterion, the development of T2I knowledge editing encounter difficulties in effectively generalizing injected knowledge. In this work, we design a T2I knowledge editing framework by comprehensively spanning on three phases: First, we curate a dataset **CAKE**, comprising paraphrase and multi-object test, to enable more fine-grained assessment on knowledge generalization. Second, we propose a novel criterion, **adaptive CLIP threshold**, to effectively filter out false successful images under the current criterion and achieve reliable editing evaluation. Finally, we introduce **MPE**, a simple but effective approach for T2I knowledge editing. Instead of tuning parameters, MPE precisely recognizes and edits the outdated part of the conditioning text-prompt to accommodate the up-to-date knowledge. A straightforward implementation of MPE (Based on in-context learning) exhibits better overall performance than previous model editors. We hope these efforts can further promote faithful evaluation of T2I knowledge editing methods.[1]

1 Introduction

Text-to-image (T2I) diffusion models have gained significant advancements in encoding real-world concepts via bridging the gap between textual descriptions and visual representations (Zhang et al., 2023a; Yang et al., 2023; Saharia et al., 2022; Rombach et al., 2022a). By pre-training on a large number of image-caption pairs, these generative models acquire statistical biases on visual concepts such as colors, objects, and personalities. For example, by inputting a text prompt “the CEO of Tesla", the model can generate a portrait of “Elon Musk". While some concepts are ageless, other encoded knowledge facts may become invalid over time (e.g., head of a state) or induce harmful social biases (e.g., implicit gender of CEO). To address this oversight, knowledge editing (Bau et al., 2020; Wang et al., 2022; Santurkar et al., 2021; Sinitsin et al., 2020; De Cao et al., 2021; Mitchell et al., 2021; Meng et al., 2022a,b) provides an efficient solution by patching undesirable model outputs without significantly altering the model’s general behavior on unrelated input.

Considering the emerging text-to-image scenario, several pioneering works have been explored for the knowledge editing of generative models (Basu et al., 2023; Arad et al., 2023; Xiong et al., 2024). These studies all borrow the idea of localized parameter updating (Meng et al., 2022a,b) from language model editing. Specifically, each fact edit is defined as a mapping from edit prompt to target prompt (e.g., "the president of the United States" $\rightarrow$ "Joe Biden") and is represented as a computed key-value vector pair. By locating this vector pair at a specific model component, such as MLP or self-attention block, one is capable of transitioning the generative model’s perception on the edit prompt to accord with up-to-date knowledge, thereby achieving knowledge editing.

However, the existing works still focus on exterior model editing, i.e., text mapping, instead of knowledge mapping and generalization reasoning. Based on an edited Stable Diffusion (Rombach et al., 2022b), we generate images by creating the input prompts that are synonymous with the fact edit and consist of multiple objects. As illustrated in Fig. 1, we observe ①**Paraphrase Generalization Failure**: Via replacing the input prompt of

[1]Our code will be made publicly available

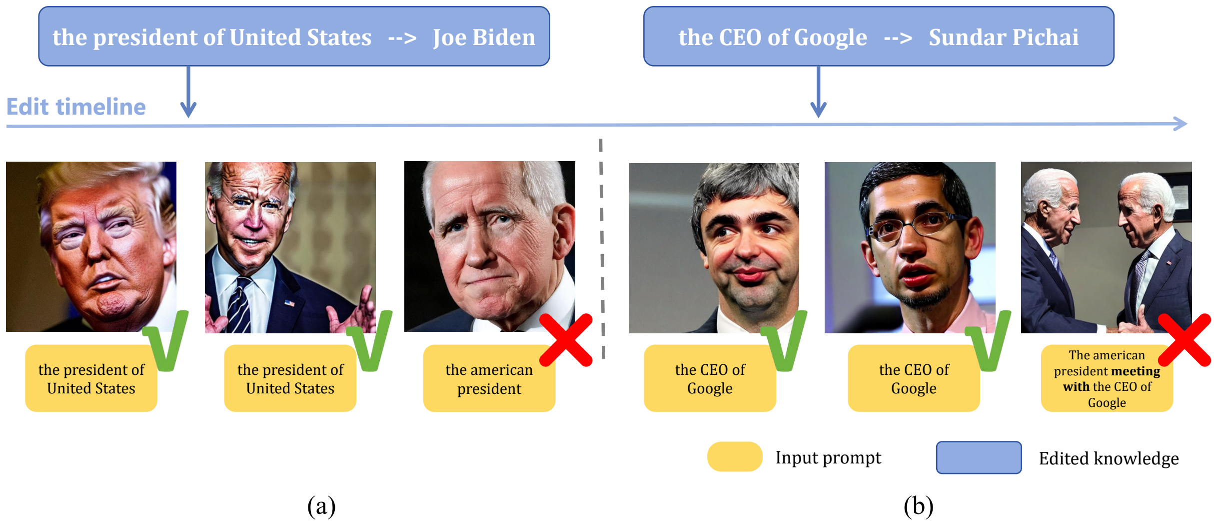

Figure 1: Illustrating the challenges in T2I knowledge editing, the **timeline** in this figure shows the order in which these images were generated: (a) Existing editing approaches often fail on paraphrases of edit prompt, such as “the American president”. We term this situation **Paraphrase Generalization Failure**. (b) The edited model struggles to deal with inputs involved with multiple edited knowledge. We refer to this case as **Compositionality Generalization Failure**.

fact edit with its paraphrase (e.g., changing "United States" to "American"), the synthetic portrait looks significantly distorted from the ground truth and distinct from the one generated by the original prompt. ②**Compositionality Generalization Failure**: When incorporating multiple edited objects within a single input prompt, the model’s generation behavior is only partially updated on a subset of fact edits. We attribute these generalization failures to superficial text mapping, where the knowledge editing lacks the reasoning flexibility to adequately comprehend various language concepts.

To effectively address how to implement knowledge mapping in generative models, which requires the edited knowledge to generalize to free and varied language inputs, we must tackle two main challenges. ❶Most of the T2I benchmark datasets (Orgad et al., 2023; Arad et al., 2023; Basu et al., 2023) used for knowledge editing do not include complex evaluation prompts comprising paraphrases and multiple edited objects. Such simple datasets hinder the development of sophisticated editing methods associated with the desired generalization capability. ❷The evaluation criterion for T2I knowledge editing are underexplored. Namely, given a synthesized image from an edited model, how can we determine whether the synthesis behavior is in line with the desired update? Previous research (Orgad et al., 2023; Arad et al., 2023) formulates the decision of editing success as a binary classification task, comparing the closeness of synthesized images to outdated and target facts. However, as shown in Fig. 1, this approach often results in false successful images that appear closer to the target facts but fail to meet the intended editing goals. Thus, a more reliable evaluation strategy is needed to advance knowledge editing efforts.

In response to these challenges, we design a comprehensive text-to-image knowledge editing framework that spans three phases: dataset construction, evaluation strategy, and editing method. First, we curate a dataset named as Counterfactual Assessment of Text-to-image Knowledge Editing (CAKE) to quantitatively assess the edited model’s capabilities in addressing the above-mentioned complex cases. In particular, CAKE introduces two new types of evaluation prompts, built from the paraphrases of edit prompt and multiple edited objects, respectively. In addition to verifying superficial text-mapping, the use of these additional evaluation prompts allows CAKE to offer a more fine-grained assessment of editing performance and insights into how well an editing method generalizes text-mapping to knowledge-mapping.

Second, to establish a reliable evaluation strategy for editing, we propose a novel criterion termed adaptive CLIP threshold. Unlike the previous criterion based on classification, this innovative criterion instead focuses on whether the synthesized image is "sufficiently" similar to the target fact. Specifically, this criterion analyzes the CLIP score distribution of ideal synthesized images and utilizes its parameter estimations to calculate a score threshold that quantifies the degree of "sufficiency". Utilizing this score threshold in decision-making can effectively filter out false successful images in editing evaluation scenarios. Our validation experiments supported by Kosmos-2, the state-of-the-art open-source vision-language model (Liu et al., 2023; Peng et al., 2023), demonstrate the superiority of the novel criterion, significantly outperforming the current criterion.

Third, rather than tuning parameters, we explore a distinctive approach to T2I knowledge editing termed **M**emory-based **P**rompt **E**diting (MPE). MPE stores all fact edits in an external memory and functions as a pre-processing module for the conditioning text prompt. Before image synthesis, MPE identifies and edits outdated parts of the input prompt to align with current knowledge. Our experiments include a simple, in-context learning-based (Brown et al., 2020) implementation of MPE. Extensive results suggest that current editing methods struggle to generalize text-mapping to desired knowledge-mapping, whereas MPE outperforms previous competitors in overall performance and applicability, demonstrating significant potential in addressing T2I knowledge editing.

2 Related Work

Text-to-image model editing. Model editing techniques focus on providing stable, targeted updates to model behavior without costly re-training. Related researches have been carried out on a variety of model architectures, such as generative adversarial networks (Bau et al., 2020; Wang et al., 2022), image classifiers (Santurkar et al., 2021) and LLMs (Meng et al., 2022a,b; Mitchell et al., 2021, 2022). (Orgad et al., 2023) formally describes T2I model editing as modifying model's generative preference for visual concepts (e.g., editing the default color of **Roses** from Red to Blue). Subsequent studies start to focus on editing factual knowledge in T2I model: Inspiring from language model editing (Meng et al., 2022a,b), ReFACT and Diff-quickfix (Arad et al., 2023; Basu et al., 2023) both encode the to-be-edited knowledge into a key-value vector pair, but place it into different model components (MLP or self-attention block). The concurrent work EMCID (Xiong et al., 2024) sequentially distributes key-value vector pairs across multiple model layers to enable massive concept editing while preserving generation quality. Unlike above methods, our proposed MPE interprets knowledge editing as prompt editing, where the model remains intact, thereby avoiding catastrophic forgetting.

3 Text-to-image Knowledge Editing

3.1 Preliminaries

Text-to-Image Diffusion Model. For our analysis, we focus specifically on T2I diffusion models. We consider a T2I diffusion model with deterministic generative processes, as described in (Song et al., 2020). This model can be expressed as $f(\mathbf{x}_T, p)$, where p represents the conditioning text prompt and $\mathbf{x}_T$ is the initial latent variable sampled from a Gaussian distribution. The function f denotes a deterministic, iterative denoising process, which outputs a real image $\mathbf{x}$.

Text-to-Image Knowledge Editing. Unlike language model editing (Meng et al., 2022a; Mitchell et al., 2021; Zhong et al., 2023; Gu et al., 2023), we define a fact edit e as a text mapping $(p_{\text{edit}} \rightarrow p_{\text{tar}})$, for example, (the U.S. president $\rightarrow$ Joe Biden). For practical applicability, we argue that the edited model should generalize the injected edits from external text mappings to internal knowledge mappings. Given an edit $e = (p_{\text{edit}} \rightarrow p_{\text{tar}})$, we formally describe the goal of T2I knowledge editing as producing an edited model f_{edit} based on f and e. The edited model f_{edit} should satisfy the following conditions:

$$\begin{aligned} \forall p \in \mathrm{Para}(p_{\text{edit}}), \quad & f_{\text{edit}}(\mathbf{x}_T, p) = f(\mathbf{x}_T, p_{\text{tar}}), \\ \forall p \notin \mathrm{Para}(p_{\text{edit}}), \quad & f_{\text{edit}}(\mathbf{x}_T, p) = f(\mathbf{x}_T, p), \end{aligned} \tag{1}$$

where $\mathrm{Para}(\cdot)$ represents the set containing all paraphrases of p_{edit}. The objective of this task requires the edited model to recognize p_{edit} in any form and map it to p_{tar} through the encoding process, which we refer to as knowledge mapping.

3.2 Counterfactual Assessment of Text-to-image Knowledge Editing

In order to faithfully assess how well the editing methods achieve knowledge mapping, we build CAKE (Counterfactual Assessment of Text-to-image Knowledge Editing) for practical and fine-grained editing evaluation. See Appendix A for dataset construction process and statistics.

Following previous work (The RoAD dataset, Arad et al., 2023), CAKE focus on counterfactual edits about figures associated with specific roles (e.g., editing **The U.S. president** $\rightarrow$ **Tim Cook**). This includes a diverse range of roles, such as

Single	Edit I: the president of the United States ->Tim Cook
Efficacy	{The president of the United States / Tim Cook}
Generality	{The president of the United States / Tim Cook} in a meeting {The president of the United States / Tim Cook} eating an apple
KgeMap	{The leader of the United States / Tim Cook} runing in the streets {The U.S. president / Tim Cook} eating strawberries
Specificity	{ flag of the United States / flag of the United States } { currency of the United States / currency of the United States }
Composite	**Edit II: the Titanic male lead ->Jeff Bezos**
Compo	{The president of United States and the Titanic male lead / Tim Cook and Jeff Bezos} hiking in the mountains {...} having a causal conversation at a coffee shop

Table 1: Part of the first entry in the CAKE dataset. All prompts are represented in $\{p_{\text{edit}}/p_{\text{tar}}\}$. During experiments, each entry undergoes top-down **alternating** editing for fair comparisons (See Appendix A for details), i.e. Edit I $\rightarrow$ evaluate {Efficacy, Generality, KgeMap, Specificity} $\rightarrow$ Edit II $\rightarrow$ evaluate {Compo}.

entrepreneurs, politicians and so on. CAKE totally contains 100 entries and each entry consists of two counterfactual edit prompts and 15 evaluation prompts, which are all represented in the form: $\{p_{\text{edit}}/p_{\text{tar}}\}$, as shown in Table 1.

After updating the knowledge expressed by the given edit prompts in a T2I model, we use different types of evaluation prompts to compute the editing performance in various dimensions:

Efficacy: Determine whether the edited model comprehends the updated text mappings.

Generality: Assess whether the edited model can flexibly utilize the updated text mappings.

Specificity: Measure how well the edited model preserves other close but unrelated concepts.

KgeMap (New): Use paraphrases to verify whether the edited model generalizes updated text mappings to knowledge mappings.

Compo (New): Evaluate the edited model's capability to apply multiple updated knowledge elements in its generative behavior simultaneously.

Evaluating in terms of the above fine-grained metrics allows CAKE to serve as a robust starting point for developing more effective and practical editing methods.

3.3 Adaptive CLIP Threshold Criterion

After updating a fact edit to a T2I model and synthesizing an image conditioned on an evaluation prompt, the critical question becomes: **How can we determine whether the synthesis aligns with the desired update?**

Previous researches (Arad et al., 2023; Orgad et al., 2023) formulate the question as a binary classification task and use the CLIP-Score $\text{CLIP}(\cdot,\cdot)$ (Radford et al., 2021; Hessel et al., 2021) to measure text-image similarity, setting the **current decision boundary** for determining editing success. However, this approach overlooks whether the synthesized image is "sufficiently" close to the target fact, leading to false positives where ineligible images are mistakenly labeled as successful (see Fig 2).

To address this, we propose an **adaptive CLIP threshold** that better aligns with the **ideal decision boundary**. By analyzing the CLIP-Score distribution of ideal images, we establish a prompt-specific threshold that quantifies "sufficiency", providing a more precise and reliable measure for evaluating edits.

To obtain the threshold, an extra warm-up stage is required before editing, as illustrated in Fig. 2. For each evaluation prompt $\{p_{\text{edit}}/p_{\text{tar}}\}$, we use the clean T2I model f conditioned on p_{tar} to generate a set of real images $\{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(n)}\}$, where $\mathbf{x}^{(i)} = f(\mathbf{x}_T^{(i)}, p_{\text{tar}})$ and $\mathbf{x}_T^{(i)}$ is the randomly sampled initial variable. These real images inherently bear sufficient similarity to the target fact p_{tar} and are thus considered ideal for post-editing generation, i.e., $f_{\text{edit}}(\mathbf{x}_T, p_{\text{edit}})$.

Next, we calculate the CLIP-Score between these ideal images and p_{tar} to form an ideal score set $S = \{s^{(1)},\ldots,s^{(n)}\}$, where $s^{(i)} = \text{CLIP}(\mathbf{x}^{(i)}, p_{\text{tar}})$. We assume the ideal score s follows a normal distribution $N(\mu,\sigma)$ and estimate its parameters $\hat{\mu}$ and $\hat{\sigma}$ using Maximum Likelihood Estimation (Pan et al., 2002):

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} s^{(i)}, \quad \hat{\sigma} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(s^{(i)}-\hat{\mu})^2}, \quad (2)$$

where $\hat{\mu}$ and $\hat{\sigma}$ are the unbiased parameter estimates for $N(\mu,\sigma)$. We define an operator $g(\hat{\mu},\hat{\sigma})$ that calculates the minimum successful similarity as the decision-making threshold, to preserve most ideal images while filtering out most unsuccessful images, as follows:

$$\text{CLIP}(f_{\text{edit}}(\mathbf{x}_T, p_{\text{edit}}), p_{\text{tar}}) \geq g(\hat{\mu},\hat{\sigma}). \quad (3)$$

Eq. (3) formulates the new criterion for editing evaluation. To determine the optimal operator $g(\hat{\mu},\hat{\sigma})$ for the knowledge editing task, we conducted a criterion validation experiment. We tested existing editing methods, TIME (Orgad et al., 2023) and ReFACT (Arad et al., 2023), on the role-editing benchmark RoAD (Arad et al., 2023) using several

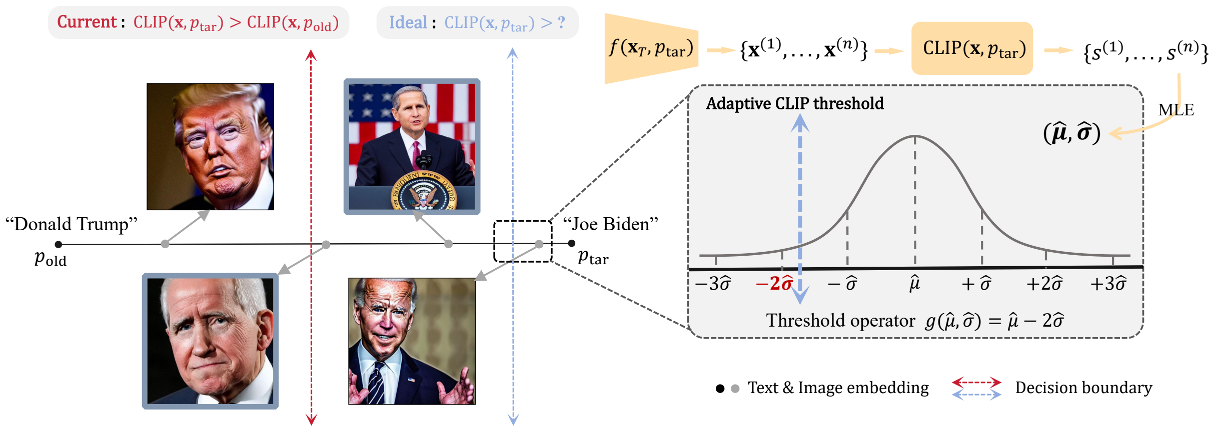

Figure 2: An editing evaluation example (p_{edit} ="the U.S. president", p_{tar} ="Joe Biden"). A closer distance between two embedding points implies higher similarity, i.e. CLIP-Score. The images with **borders** are false successful images under the current criterion. For each evaluation prompt, the adaptive CLIP threshold precisely approximates the **ideal decision boundary** and effectively filters out the false successful images.

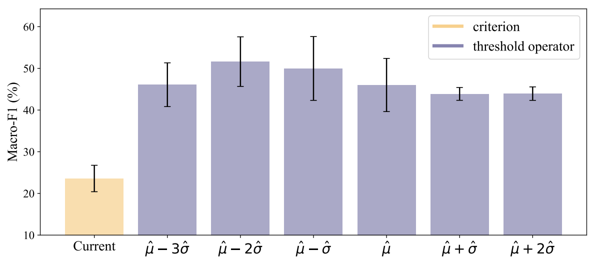

Figure 3: The Macro-F1 of different criterion(threshold operators). **Current** refers to the current criterion.

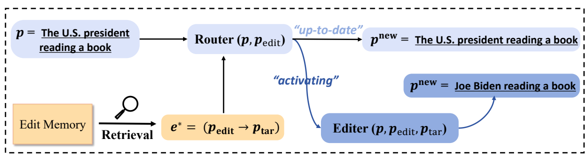

Figure 4: The basic workflow of MPE.

operator choices (e.g., $\hat{\mu} - 2\hat{\sigma}$) to make evaluation decisions. Additionally, we selected Kosmos-2 (Peng et al., 2023), the best-performing open-source vision-language model for the **Celebrity Recognition** task (Liu et al., 2023), as the pseudo-label generator (see Appendix B for the pseudo-label generation process)[2]. Fig. 3 presents the Macro-F1 performance of various operator choices and the current classification-based criterion. The results demonstrate that $\hat{\mu} - 2\hat{\sigma}$ is the most effective choice among the candidate operators. Furthermore, the adaptive CLIP threshold consistently outperforms the current criterion, indicating its reliability as an evaluation scheme. In later experiments, we set threshold operator $g(\hat{\mu}, \hat{\sigma}) = \hat{\mu} - 2\hat{\sigma}$.

3.4 MPE: A Proposal for Text-to-Image Knowledge Editing

In this section, we propose a simple and effective scheme for T2I knowledge editing, MPE (**M**emory-based **P**rompt **E**diting).

Workflow. Unlike previous parameter-update methods, when receiving a fact edit ($p_{edit} \rightarrow p_{tar}$), MPE keeps the T2I model frozen and serves as a pre-processing module for the conditioning text prompt p, as follows:

$$f_{edit}(\mathbf{x}_T, p) = f(\mathbf{x}_T, \mathrm{MPE}(p, p_{edit}, p_{tar})). \quad (4)$$

Towards the task objective defined in Sec 3.1, the expected output of MPE should be either p_{tar} or p, depending on whether $\mathrm{Para}(p_{edit})$ contains p itself or any sub-sequence of p (e.g., the ideal output of "The U.S. president reading a book" should be "Joe Biden reading a book").

In particular, MPE consists of two components: Router and Editer. 1) The Router takes p and p_{edit} as input and detects whether the p contains any paraphrases from $\mathrm{Para}(p_{edit})$. If so, it sends an "activating" signal to the Editer, which implies the generating behavior on p of the clean model f has been outdated. 2) If receiving the signal, the Editer would precisely recognize the outdated part (any form of the p_{edit}) of the input prompt p and then replace it with the p_{tar}. Depending on MPE, the text prompt can adaptively fuse with edited knowledge, thereby altering the T2I model's generation behavior in a targeted way, as shown in Fig 4.

Multiple editing. Real-world scenarios generally involve a vast pool of knowledge updates. To operate in practical applications, MPE adopts a "Mem-

[2]The ability of GPT-4v to perform person identification has been officially prohibited. Thus, Kosmos-2 was chosen.

Method	Score	Efficacy	Generality	KgeMap	Compo	Specificity	FID (↓)	CLIP
Base	0.00	00.00%±0.00	03.09%±0.93	03.10%±0.67	01.73%±0.66	**96.90%**±1.53	33.41	0.426
TIME	11.4	03.50%±0.92	12.68%±1.73	10.37%±1.62	04.80%±1.17	<u>85.80</u>%±3.09	31.94	0.421
ReFACT	35.2	33.70%±6.18	42.46%±5.51	34.10%±4.48	35.73%±4.87	31.19%±2.09	33.38	0.426
EMCID	41.9	82.60%±8.82	48.48%±4.73	39.43%±2.89	40.83%±6.93	19.97%±1.50	32.65	0.426
MPE	**77.2**	**<u>94.40</u>**%±2.73	**<u>88.84</u>**%±4.52	**<u>63.07</u>**%±2.52	**<u>72.70</u>**%±3.35	71.20%±1.87	33.41	0.426

Table 2: Quantitative evaluation results on CAKE. Best results are marked with **bold**. Best results among editing methods are marked with <u>underline</u>. **FID** refers to FID-5K, **CLIP** refers to the average CLIP Score.

ory + Retrieval" strategy (Mitchell et al., 2022; Gu et al., 2023; Song et al., 2024) and introduces an additional Retriever component. Specifically, when receiving multiple edits $\{e^{(1)}, \ldots, e^{(n)}\}$, MPE stores all edits in an external memory and embeds their $p^{(i)}_{\text{edit}}$ by the Retriever to construct a retrieval index. Then for each input prompt p, the retrieval index returns the key edit e^* that is the most relevant (i.e., closest in the embedding space) to p, and sends them together to the Router for prompt editing. The complete workflow of MPE is described in Appendix C.

Implementation. The Router and the Editer can be instantiated using various schemes, such as fine-tuning a pre-trained text classification model (Sanh et al., 2019; Devlin et al., 2018) for the Router and a Seq2Seq model (Lewis et al., 2019; Raffel et al., 2020) for the Editer. In this paper, we consider a lightweight, in-context learning-based implementation: We deploy the pre-trained Contriever model (Izacard et al., 2022) locally as the Retriever component and teach the GPT-3.5-turbo API (Ouyang et al., 2022) to work as both the Router and the Editer simultaneously, by our manually designed demonstrations (i.e., input-label pairs). The concrete prompts used are detailed in Appendix D.

4 Experiments

4.1 Experimental Setup

In this paper, we investigate both single-editing (updating edits from a single entry at a time) and multiple-editing (updating edits from multiple entries at a time) scenarios for comprehensive assessment. All experiments are conducted using the Stable Diffusion v1-4 model (Rombach et al., 2022b).

Dataset. In addition to the newly constructed CAKE, we include the knowledge editing dataset RoAD (Arad et al., 2023) and the preference editing TIME Dataset (Orgad et al., 2023) in our experiments. The TIME Dataset contains 147 variations about visual concepts (e.g., changing the default color of <u>Roses</u> from Red to Blue) to assess the performance in editing generative preference.

Baseline. Except for the unreleased Diff-quickfix (Basu et al., 2023), we experiment with all available T2I knowledge editing baselines, including TIME (Orgad et al., 2023), ReFACT (Arad et al., 2023), and EMCID (Xiong et al., 2024). TIME targets at modifying generative preferences and cannot be directly applied to RoAD and CAKE due to the incompatible input format. So we implement an adaptation version of TIME that has been empirically demonstrated to be the most effective version in knowledge editing scenarios (Arad et al., 2023). Following prior settings, we include a special case, <u>Base</u>, in our single-editing experiments. For each evaluation prompt $\{p_{\text{edit}}/p_{\text{tar}}\}$, <u>Base</u> refers to directly inputting p_{edit} into the unedited model f for generation, serving as a reference baseline.

Metric. We introduce the metrics we considered in Section 3.2. We evaluate editing performance in terms of Efficacy, Generality, Specificity, KgeMap and Compo. Among them, KgeMap and Compo are only available for the CAKE dataset. We use our proposed adaptive CLIP threshold as the evaluation criterion. After editing, an evaluation prompt $\{p_{\text{edit}}/p_{\text{tar}}\}$ is considered successful if the synthesized image $\mathbf{x}$ conditioned on p_{edit} satisfies $\text{CLIP}(\mathbf{x}, p_{\text{tar}}) \geq \hat{\mu} - 2\hat{\sigma}$. Then each metric is computed as the ratio of successful evaluation prompts to the total number of corresponding evaluation prompts. We also calculate the geometric mean of all the aforementioned metrics as <u>Score</u> to characterize the overall performance. To evaluate the general image quality, we report the FID-5K (Heusel et al., 2017) and the average CLIP score (Radford et al., 2021) based on a randomly selected 5,000 image-caption pairs from the MS-COCO validation dataset (Lin et al., 2014). We use Laion's ViT-G/14 (Cherti et al., 2023), the best open-source CLIP model, to conduct all CLIP Score calculation.

Setting. For each evaluation prompt $\{p_{\text{edit}}/p_{\text{tar}}\}$: Before editing, we need an extra warm-up stage to

Dataset	Method	Score	Efficacy	Generality	Specificity	FID(↓)	CLIP
RoAD	Base	15.8	02.89%±1.66	14.11%±1.10	**95.98**%±1.26	33.41	0.426
	TIME	44.6	28.78%±3.12	37.42%±1.59	82.60%±3.39	31.60	0.422
	ReFACT	57.1	39.11%±4.44	53.53%±2.72	<u>88.87</u>%±1.10	33.36	0.426
	EMCID	78.9	85.00%±4.07	69.18%±3.06	83.51%±1.58	33.09	0.426
	MPE	<u>**87.6**</u>	<u>**90.89**</u>%±3.58	<u>**89.31**</u>%±2.36	82.69%±1.41	33.41	0.426
TIME Dataset	Base	49.9	25.77%±3.09	50.85%±2.06	**95.15**%±1.99	33.41	0.426
	TIME	81.8	84.52%±4.46	79.06%±2.43	82.02%±3.34	31.78	0.423
	ReFACT	73.7	65.38%±4.26	70.87%±2.32	<u>86.31</u>%±1.36	33.39	0.426
	EMCID	79.5	88.65%±3.12	80.54%±2.04	70.31%±1.94	33.18	0.426
	MPE	<u>**86.4**</u>	<u>**97.02**</u>%±1.63	<u>**91.58**</u>%±1.12	72.65%±1.73	33.41	0.426

Table 3: Quantitative evaluation results on RoAD and TIME Dataset. Best results are marked with **bold**. Best results among editing methods are marked with <u>underline</u>.

Dataset	Method	#1	#10	#25	#50	#All
CAKE	TIME	11.36%	00.00%(0%)	00.00%(0%)	00.12%(1%)	00.00%(0%)
	ReFACT	35.24%	27.76%(78%)	23.84%(67%)	21.62%(61%)	20.15%(57%)
	EMCID	41.87%	33.54%(80%)	30.42%(73%)	29.27%(70%)	25.85%(62%)
	MPE	**77.18%**	**77.17%**(99%)	**75.54%**(97%)	**75.93%**(98%)	**74.83%**(96%)

Table 4: The metric <u>Score</u> in multiple editing experiments on CAKE is reported here to characterize the trend in overall editing performance. The **(# num)** refers to the size of edit batch. The **(percent %)** indicates the percentage to which the editing methods preserve the single-editing performance **(# 1)**. Best results are marked with **bold**.

calculate the adaptive CLIP threshold over 50 random seeds; After editing, we generate synthesized images conditioned on p_{edit} over 10 random seeds to obtain the stable editing performance. Various seeds correspond to different initial variables $\mathbf{x}_T$. All experiments are conducted on NVIDIA A40s and take about 15 GPU hours to finish one setting.

4.2 Single Editing Results

Table 2,3 presents our single-editing results. We observe that our proposed **MPE** demonstrates superior overall performance compared to other baselines across all datasets, especially in the knowledge editing task (CAKE, RoAD), underscoring its potential for further development.

The experimental results on CAKE are consistent with our early findings: current editing methods struggle to generalize text-mapping to desired knowledge-mapping, as evidenced by their performance degradation in both the KgeMap and Compo metrics. This poses significant challenges for future research endeavors.

The **TIME** method, originally designed for editing generative preferences, fails catastrophically on CAKE and thus proves inadequate for updating factual knowledge within the diffusion model. However, its exceptional and well-balanced performance on its initial task (TIME Dataset) remains noteworthy. Considering its low computational cost and rapid editing speed, TIME presents itself as a strong alternative for preference editing.

Quantitatively, the overall performance of **ReFACT** is relatively low, only surpassing TIME in knowledge editing tasks. Meanwhile, as illustrated by the qualitative examples in Fig. 5, the synthesis behaviors of the ReFACT-edited model progress in the desired direction but ultimately fail. These "plausible" images can be effectively filtered out using the adaptive CLIP threshold.

EMCID exhibits superior performance among parameter-update editing methods. On RoAD, EMCID distinguishes itself by demonstrating excellent performance across all considered metrics; On CAKE, EMCID is able to generate images that better match the editing goal than ReFACT (See Fig. 5). However, the weak Specificity in Table 2 indicates that EMCID struggles to limit the editing scope, encountering difficulties in correctly generating close but unrelated concepts after editing.

Interestingly, compared to the superior overall performance, MPE does not excel in Specificity. We attribute this to the drawbacks of prompt editing: once the pre-processing module make a mistake, the revised prompt could be totally unrelated to the original input (e.g., flag of the United States $\rightarrow$ Tim Cook). Fortunately, we later observe that

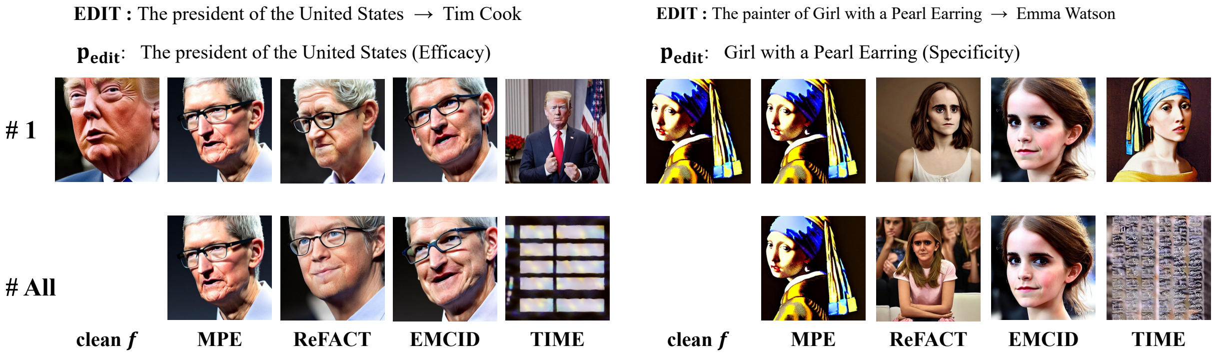

Figure 5: The qualitative examples from the CAKE dataset. The **(# num)** refers to the size of edit batch.

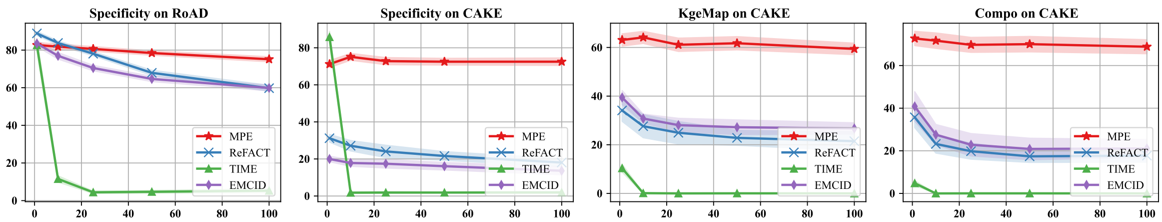

Figure 6: The performance curves of various metrics across multiple editing experiments are depicted. The horizontal axis denotes the size of the edit batches, while the shaded areas indicate the standard deviation.

when facing high edit volumes, the Specificity of MPE exhibits excellent robustness, potentially compensating for the identified shortcoming.

4.3 Multiple Editing Results

We conducted multiple editing experiments to simulate real-world scenarios. We group entries into edit batches of size k, where k takes values from $\{1, 10, 25, 50, \text{all}\}$. Then for each batch, we injected all fact edits within it into the clean model simultaneously and evaluated the performance on all associated evaluation prompts.

Table 4, Fig. 6 present the related results. We first investigate the changing trend in overall editing performance: Except MPE, other (parameter-update) editing methods have suffered considerable performance degradation – TIME completely lost its editing ability; The performance of ReFACT under (**#All**) has also declined to nearly half of its single-editing performance; EMCID exhibits better robustness to larger edit volumes, benefited from its distributed editing strategy, but is still significantly inferior to MPE. Utilizing a proficient external retriever, MPE demonstrates outstanding performance retention (96%) under (**#All**). Besides, qualitative examples in Fig. 5 show that 1) TIME frequently generates meaningless pure noise under multiple editing, which reveals the loss in generating ability caused by parameter updates; 2) ReFACT and EMCID maintain image quality well, suggesting that the MLPs in the text encoder might be a better updating location for knowledge editing.

We then focus on some specific metrics. The curves in Fig. 6 show that MPE owns remarkable robustness to multiple editing, which potentially compensates its weaknesses in Specificity. Conversely, the robustness of ReFACT and EMCID to multiple editing seems less than ideal: They both experience relatively large performance degradation across all metrics. We hope these results can act as a call to the community to develop more practical and effective editing methods. More quantitative and qualitative results are provided in Appendix E.

5 Conclusion

In this work, we aim to establish a reliable evaluation paradigm for T2I knowledge editing. Specifically, we curate a dataset named CAKE, comprising fine-grained metrics to validate knowledge generalization. We then develop an innovative criterion, the adaptive CLIP threshold, to approximate the ideal decision boundary, effectively filtering out false successful images in evaluation scenarios. Additionally, by transferring the editing impact from the parameter space to the input space, we design a distinctive approach, MPE, to achieve T2I knowledge editing. Extensive results have demonstrated the limitations of current editing methods and the further potential of MPE.

Limitations

The limitations of our work are as follows:

1. Similar to previous datasets, our curated CAKE focuses on figure editing pertaining to specific roles. To maintain the quality of evaluation prompts, the scale of CAKE is kept small, comprising only 100 edits and 1,500 evaluation prompts. We suggest that future research should aim to construct a larger and more diverse knowledge editing dataset to achieve more reliable evaluations.

2. Our experiments only involve a straightforward, API-based implementation of our proposed MPE. The further potential of MPE in real applications is under-explored because the call of OpenAI API leads to inevitable financial costs. In future work, we will experiment with more economical schemes of MPE as stated in Sec. 3.4.

3. Memory-based editing allows for lossless editing of models and thus distinguishes itself among editing techniques. However, its vulnerability to attacks such as memory injection poses significant risks in production environments. Therefore, this approach requires robust security measures to mitigate these risks effectively in real-world scenarios.

Ethics Statement

We curate a counterfactual editing dataset named CAKE, which includes world-renowned roles and identifiable figures. During the dataset construction process, we faithfully adhere to privacy regulations and collect publicly available information from the internet. We randomly assign counterfactual relations between specific roles and figures. On behalf of all authors, we declare that these counterfactual relations are exclusively intended for research purposes and carry no implications for the real world. We have manually ensured that the finished dataset does not contain any potentially offensive content.

References

Dana Arad, Hadas Orgad, and Yonatan Belinkov. 2023. Refact: Updating text-to-image models by editing the text encoder. *arXiv preprint arXiv:2306.00738*.

Samyadeep Basu, Nanxuan Zhao, Vlad I Morariu, Soheil Feizi, and Varun Manjunatha. 2023. Localizing and editing knowledge in text-to-image generative models. In *The Twelfth International Conference on Learning Representations*.

David Bau, Steven Liu, Tongzhou Wang, Jun-Yan Zhu, and Antonio Torralba. 2020. Rewriting a deep generative model. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 351–369. Springer.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. 2023. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. *arXiv preprint arXiv:2104.08164*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Hengrui Gu, Kaixiong Zhou, Xiaotian Han, Ninghao Liu, Ruobing Wang, and Xin Wang. 2023. Pokemqa: Programmable knowledge editing for multi-hop question answering. *arXiv preprint arXiv:2312.15194*.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2021. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Preprint*, arXiv:2112.09118.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2022. Pretrained language models for text generation: A survey. *arXiv preprint arXiv:2201.05273*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. 2023. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2021. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. 2022. Memory-based model editing at scale. In *International Conference on Machine Learning*, pages 15817–15831. PMLR.

Hadas Orgad, Bahjat Kawar, and Yonatan Belinkov. 2023. Editing implicit assumptions in text-to-image diffusion models. *arXiv preprint arXiv:2303.08084*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Jian-Xin Pan, Kai-Tai Fang, Jian-Xin Pan, and Kai-Tai Fang. 2002. Maximum likelihood estimation. *Growth curve models and statistical diagnostics*, pages 77–158.

Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. 2023. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022a. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022b. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Shibani Santurkar, Dimitris Tsipras, Mahalaxmi Elango, David Bau, Antonio Torralba, and Aleksander Madry. 2021. Editing a classifier by rewriting its prediction rules. *Advances in Neural Information Processing Systems*, 34:23359–23373.

Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitriy Pyrkin, Sergei Popov, and Artem Babenko. 2020. Editable neural networks. *arXiv preprint arXiv:2004.00345*.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Xiaoshuai Song, Zhengyang Wang, Keqing He, Guanting Dong, Jinxu Zhao, and Weiran Xu. 2024. Knowledge editing on black-box large language models. *arXiv preprint arXiv:2402.08631*.

Sheng-Yu Wang, David Bau, and Jun-Yan Zhu. 2022. Rewriting geometric rules of a gan. *ACM Transactions on Graphics (TOG)*, 41(4):1–16.

Tianwei Xiong, Yue Wu, Enze Xie, Yue Wu, Zhenguo Li, and Xihui Liu. 2024. Editing massive concepts in text-to-image diffusion models. *arXiv preprint arXiv:2403.13807*.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. 2023. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4):1–39.

Chenshuang Zhang, Chaoning Zhang, Mengchun Zhang, and In So Kweon. 2023a. Text-to-image diffusion model in generative ai: A survey. *arXiv preprint arXiv:2303.07909*.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2023b. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Computing Surveys*, 56(3):1–37.

Zexuan Zhong, Zhengxuan Wu, Christopher D Manning, Christopher Potts, and Danqi Chen. 2023. Mquake: Assessing knowledge editing in language models via multi-hop questions. *arXiv preprint arXiv:2305.14795*.

A Statistics and Construction Details of CAKE

Statistics. CAKE comprises 100 different edits and 1,500 evaluation prompts. Each entry includes two edits (**Edit I**, **Edit II**) along with the corresponding evaluation prompts for performance assessment: 1 Efficacy prompt, 5 Generality prompts, 3 Specificity prompts, 3 KgeMap prompts, 3 Compo prompts.

Construction Details. Given the powerful text generation capabilities of LLMs (Li et al., 2022; Zhang et al., 2023b), we utilize ChatGPT to automatically gather candidate edit prompts p_{edit} and target prompts p_{tar} to form fact edits. Specifically, we prompt ChatGPT to:

i) list the top-20 influential individuals across various fields of our time (e.g., Jeff Bezos, Tim Cook) to create a candidate target set $\mathcal{O} = \left\{p_{\text{tar}}^{(1)}, \ldots, p_{\text{tar}}^{(20)}\right\}$. We manually verified their correct generation of Stable Diffusion v1-4 (Rombach et al., 2022b), the text-to-image diffusion model we study.

ii) generate 10 roles in different categories (e.g., the CEO of Microsoft).

iii) for each role, leverage in-context learning (Brown et al., 2020) to automatically produce 9 additional roles in same category (e.g., the CEO of Tesla, the CEO of IBM) to gather a candidate edit prompt set $\left\{p_{\text{edit}}^{(1)}, \ldots, p_{\text{edit}}^{(100)}\right\}$.

Then for each existing p_{edit}, we randomly assign a target prompt in $\mathcal{O}$ to it and construct a counterfactual text-mapping (edit) set $\mathcal{E} = \{e_1, \ldots, e_{100}\}$. We refer to each existing edit as **Edit I** and build evaluation prompts for them to compose the complete entry. In particular, for all metrics except Specificity, we fill the p_{edit}/p_{tar} pairs into natural language templates (e.g., _ eating an apple) to form evaluation prompts. In the case of Specificity, we manually design evaluation prompts (e.g., Tesla logo) inquiring about other knowledge related to the entities (e.g., Tesla) in p_{edit}.

We then further augment the existing dataset by introducing **Edit II**: For each entry, we supplement it with a randomly sampled edit ($p'_{\text{edit}} \rightarrow p'_{\text{tar}}$) from the rest of single-edit part that satisfies $p_{\text{tar}} \neq p'_{\text{tar}}$. We term the newer edit as **Edit II**.

Finally, each candidate entries was independently reviewed by us in terms of grammar and semantic logic. The outcome of this meticulous process was the CAKE dataset comprising 100 entries.

The top-down alternating editing. The editing and evaluation order of CAKE is slightly different from other editing datasets. After updating the **Edit I** to the T2I model, we first finish the generations on evaluation prompts of { Efficacy, Generality, Specificity, KgeMap}. Afterwards, we directly insert the **Edit II** into the current, edited model and finally compute the last metric { Compo}. By following the top-down alternating editing, we test the Compositionality property and can precisely compute the editing performance of T2I model with only one newer edit, aligning with other editing datasets.

B Detailed process of the Criterion Validation Experiments

To and the most effective threshold operator and validate the superiority of our proposed adaptive CLIP threshold, we leverage the Kosmos-2 (Peng et al., 2023) as the pseudo-label generator, enabling the automatic criterion evaluation. Specifically, Kosmos-2 is prompted to conduct **celebrity recognition** task (Kosmos-2 is the best open-source VLM on this task according to (Liu et al., 2023)).

Following previous settings, we adopt the zero-shot context for Kosmos-2 to execute the visual question answering task. For each synthesised image from existing editing methods, Kosmos-2 is taught to answer the question "Who is the person in this image?" with subsequent four options. One of these options corresponds to the target figure after editing, while the others are randomly selected from a pool of candidate celebrities. A synthesised image is labeled as "successful" only if Kosmos-2 selects the correct option or directly outputs the

Several demonstrations in MPE's in-context prompt
1. Input: The spokesman of United Nations giving a speech source concept: The chief trainer of Inter Miami. target concept: David Beckham. Does the entity specified by source concept appeared in the Input: No. Output: The spokesman of United Nations giving a speech
2. Input: The lead singer of Nightwish standing on the stage source concept: The lead singer of Nightwish. target concept: Elvis Presley. Does the entity specified by source concept appeared in the Input: Yes. Output: Elvis Presley standing on the stage
3. Input: Kylian Mbappe and Kanye West celebrating Christmas together source concept: The chief scientist at NASA. target concept: Boris Johnson. Does the entity specified by source concept appeared in the Input: No. Output: Kylian Mbappe and Kanye West celebrating Christmas together

Table 5: Here are several demonstrations from MPE's in-context prompt. When the language model answers the question, 'Does the entity specified by the source concept appear in the input?', it functions as the Router. When the language model generates the final output, it functions as the Editer.

name of the target figure.

C Overall Algorithm of MPE

In Sec 3.4, we present the basic workflow of MPE. However, in real applications, when receiving a text prompt p, we don't actually know how many fact edits it's associated with. So, to accommodate this problem, we leverage the Router R to determine whether the editing process should be terminated. The specific algorithm is in Alg. 1.

Algorithm 1 Overall Workflow of MPE.

Input: edit memory $\mathcal{M} = \{e^{(1)}, \ldots, e^{(n)}\}$, router R, editer E, retriever $\mathrm{Retrieval}()$, input text prompt p

```
/* Editing in the loop */
for M ≠ ∅ do
    e* = Retrieval(M, p)
    M = M \ {e*}
    if R(p, p*_edit) ≠ "Activating" then
        return p
    end if
    p = E(p, p*_edit, p*_tar)
end for
```

D Prompts used for In-context Learning

We present several demonstrations from MPE's in-context prompt in Table 5 to illustrate the working mechanism of in-context learning-based MPE implementation.

E More Quantitative and Qualitative Results

The performance curves of editing methods in terms of { Efficacy, Generality} are presented in Fig. 7.

The results of the metric Score on RoAD in multiple-editing are shown in Table 6.

Additional qualitative examples in metrics { KgeMap, Compo } are provided in Fig. 8

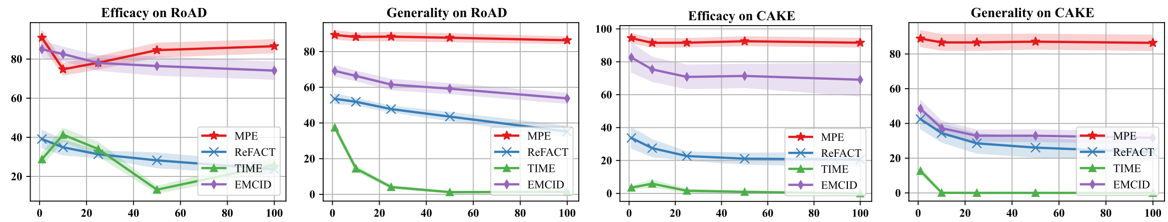

Figure 7: The performance curves of various metrics across multiple editing experiments are depicted. The horizontal axis denotes the size of the edit batches, while the shaded areas indicate the standard deviation.

Dataset	Method	#1	#10	#25	#50	#All
RoAD	TIME	44.64%	19.03%(42%)	8.52%(19%)	04.25%(9%)	05.80%(12%)
	ReFACT	57.09%	53.33%(93%)	48.89%(85%)	43.70%(76%)	36.78%(64%)
	EMCID	78.89%	74.99%(95%)	69.67%(88%)	66.40%(84%)	62.03%(78%)
	MPE	**87.56%**	**81.42%**(92%)	**82.26%**(93%)	**83.50%**(95%)	**82.49%**(94%)

Table 6: The metric Score in multiple editing experiments on RoAD is reported here to characterize the trend in overall editing performance. The **(# num)** refers to the size of edit batch. The **(percent %)** indicates the percentage to which the editing methods preserve the single-editing performance **(# 1)**. Best results are marked with **bold**.

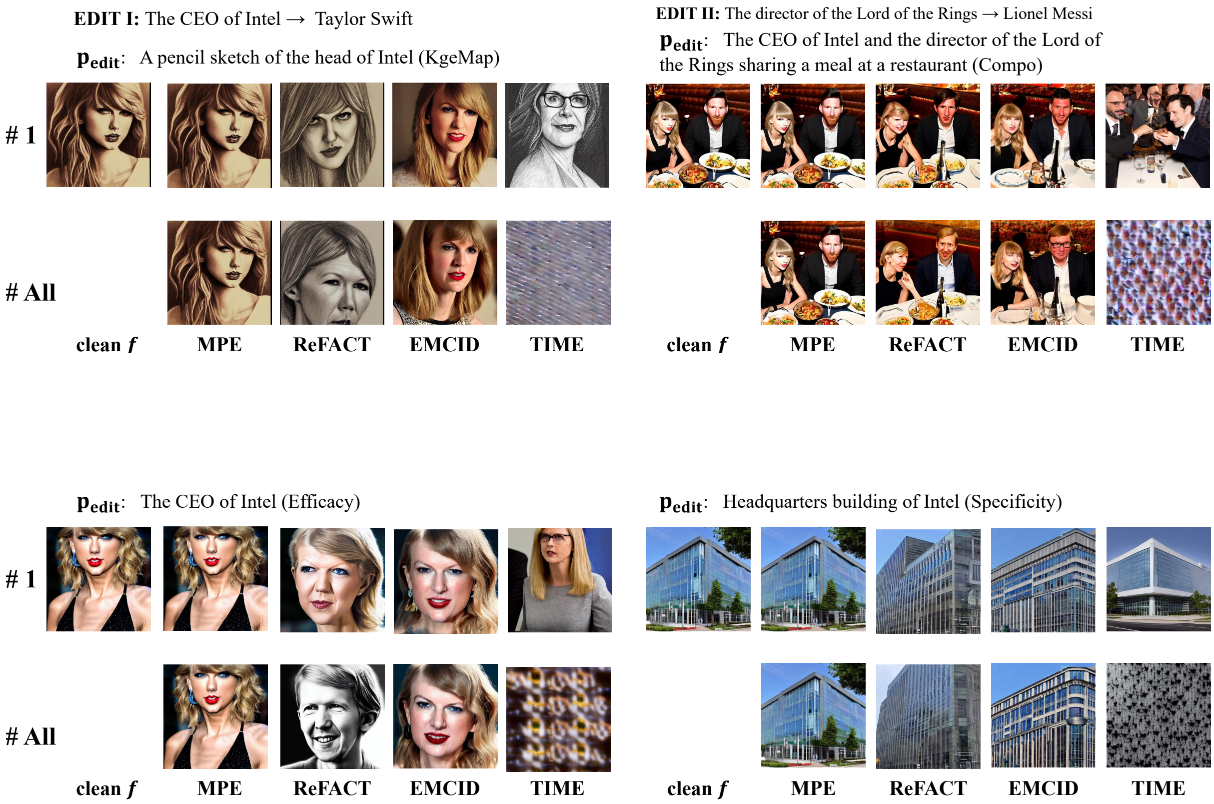

Figure 8: The qualitative examples from the CAKE dataset. The **(# num)** refers to the size of edit batch.