VISUALS MATTER: BENCHMARKING AND ENHANCING MULTI-MODAL KNOWLEDGE EDITING UNDER VISUAL DISTRACTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal knowledge editing (MMKE) has emerged as a critical approach for updating outdated knowledge in Vision-Language Models (VLMs). For MMKE, the knowledge must be grounded in visual cues that identify the target subject, so that the knowledge can transfer to any image of that subject while unrelated knowledge is preserved. However, real-world images often contain irrelevant visual elements, which we term as *visual distractions*. Existing MMKE benchmarks do not systematically examine their effects, leaving an open question: *how do visual distractions affect MMKE?* We introduce DISTRACTEDIT, a controlled, semi-synthetic benchmark that evaluates the same knowledge updates across increasingly complex visual scenarios: Isolated Entity, Contextual Background, and Multi-Subject Interference. By preserving the target subject's visual details under different scenarios, DISTRACTEDIT enables controlled comparisons that isolate the effect of distractions on editing performance. Our evaluation across MMKE methods and VLMs reveals that visual distractions undermine editing performance, particularly the generalization of edited knowledge to unseen images, and the degradation degree varies across methods, models, and distraction types. We further propose SACO, an agent-based method that refines visual inputs by mitigating distractions before knowledge editing, following a *See, Analyze, Clean, and Optimize* strategy. Our results demonstrate that mitigating visual distractions improves the performance, highlighting visual refinement as a promising direction. We present SACO as a starting point and anticipate that DISTRACTEDIT and SACO will facilitate future research on MMKE under visual distractions.

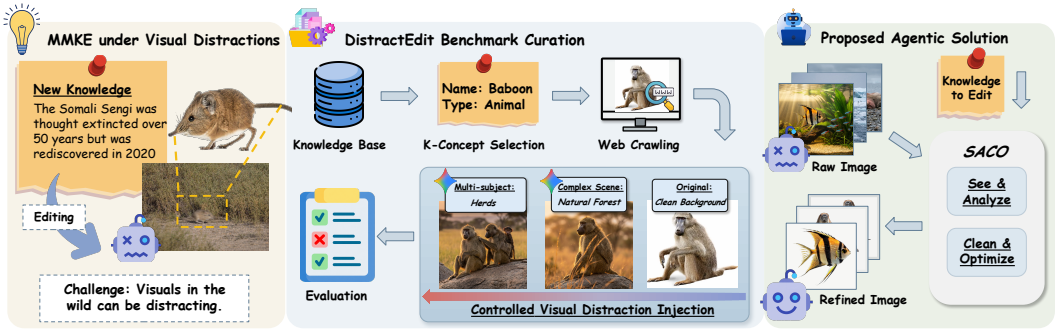

Figure 1: Overview. The left part highlights the challenge where visual distractions in the real-world photo lead to fragile knowledge editing. The middle panel illustrates how the curation of DISTRACTEDIT injects realistic visual distractions in a controlled way, so that images of the same subject mainly differ in their distractions. The right panel presents our proposed agentic solution, SACO, which refines images into distraction-free images to improve the MMKE performance.

1 INTRODUCTION

In a rapidly evolving world, the knowledge in Vision Language Models (VLMs) can quickly become outdated or incorrect (Dosovitskiy et al., 2020; Brown et al., 2020; Zhao et al., 2023; Liu et al., 2023). Knowledge editing (KE) (Wang et al., 2024d; Zhang et al., 2024c) has emerged as a remedy

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

to update specific knowledge without affecting unrelated content, avoiding frequent retraining of these massive models. Most existing KE methods either modify the model's parameters (Fang et al., 2024), or preserve parameters and update the knowledge in context (Zheng et al., 2023) or retrieve from an additional memory (Hartvigsen et al., 2023). Both families are widely used and apply to different needs. Applying KE to VLMs (Cheng et al., 2023; Huang et al., 2024), referred to as Multi-modal KE (MMKE), introduces a new dimension of complexity where the target knowledge must be grounded (Wang et al., 2024d). A successful MMKE update should bind updated knowledge to visual cues that identify the target subject, such that the edited VLM can generalize and recall the updated knowledge with any image of that subject, while leaving the knowledge about unrelated visual content unchanged (Chen et al., 2025b; Li et al., 2026; Das & Jena, 2026). This brings new challenges to the editing process, as images, particularly real-world images, often contain far more visual elements than those relevant to the knowledge being edited.

Consider the "Somali Sengi" in Fig. 1. The species *was* considered extinct until a recently captured photo proved its *rediscovery* (Garcia, 2021). However, the Sengi only occupies a small portion of the photo. Most of the contents, such as the gravel, terrain, and vegetation, are only *weakly* associated with the target knowledge. We refer to such factors, e.g. irrelevant backgrounds or other subjects that resemble or relate to the subject, as *visual distractions*. Crucially, these distractions can mislead the editing process about where the new knowledge belongs. The rediscovery knowledge can be associated with the terrain around the Sengi rather than only with the Sengi itself; the gravel further blurs the animal's shape, confusing which cues identify the subject in the image. The resulting update will be fragile. The knowledge cannot be recalled and fails to transfer to other images that contain the same subject. Although recent MMKE datasets and benchmarks have progressed toward more fine-grained and realistic knowledge updates (Cheng et al., 2023; Zeng et al., 2025; Du et al., 2025; Fu et al., 2026), they are primarily designed to evaluate what knowledge is edited rather than how the visual distractions in the editing image affect the edit. In other words, they leave an open question: *how would visual distractions affect MMKE?*

To answer this question, we develop DISTRACTEDIT to systematically evaluate and analyze MMKE under varying forms of visual distractions. The core design of DISTRACTEDIT is the controlled comparison. For each subject, the same knowledge update is performed under different visual scenarios. In this way, the differences in editing performance can be attributed to visual distractions. We consider three progressively more complex visual scenarios. First, Isolated Entity (Original) establishes a base scenario where the visual subject dominates the center of the image with minimal distractions. The second scenario, Contextual Background (Complex Scene), places the subject in a realistic, cluttered background. Because the background introduces no additional knowledge concept, its content is purely distracting. Finally, the third scenario, Multi-Subject Interference (Multi-Subj), contains visually similar or semantically related subjects. To isolate the effects of visual distractions, the images of the same subject across different visual scenarios should mainly differ in visual distractions. However, real-world images of the same subject vary in many factors, such as appearance, resolution, and image quality, and thus do not provide controlled comparisons. Therefore, we adopt a semi-synthetic curation framework. Starting from real-world Isolated Entity images, we synthesize the other two scenarios by progressively introducing realistic visual distractions using VLM-guided image editing with Gemini and Nano Banana (Google DeepMind, 2026), followed by human verification that filters out low-quality images. Across scenarios, the resulting images preserve each subject's visual details and differ mainly in their distractions.

We extensively evaluate eight KE methods on two representative VLM backbones, InternVL-3.5 8B (Wang et al., 2025) and Blip2-OPT 2.7B (Li et al., 2023). Our results reveal systematic failures in existing KE paradigms when facing visual distractions. First, we note that when visual distractions are present, existing methods often result in *fragile* editing: *the edited model can only recognize the edited knowledge when seeing the specific visual being used for editing*, leading to a notable degradation in the generalizability of the new knowledge. Moreover, the degradation degree depends on multiple factors, such as VLM architectures, distraction types, and the editing mechanism itself.

Given the pivotal role of visual distractions revealed by DISTRACTEDIT, an intuitive approach is to remove the distractions from visual inputs before KE to improve current MMKE methods. As a first step in this direction, we propose SACO, a VLM-based agent to See, Analyze, Clean, and Optimize visual inputs. Given only the raw image and editing request, SACO first analyzes knowledge-relevant visual cues and potential visual distractions. Based on the analysis, SACO formulates instructions for image editing tools to refine the visual input by removing distracting

elements while preserving the core knowledge concept. Empirically, we show that SACO serves as a plug-and-play module that mitigates the vulnerability of existing KE methods to visual distractions, indicating that mitigating visual distraction is necessary. This also opens up several directions for further improvement, including more advanced agentic structures, adaptive visual refinement, and editing methods that are natively robust to complex visual contexts. We present SACO as a starting point for future research.

To facilitate future research, we will release our benchmark DISTRACTEDIT and SACO, and invite the community to benchmark new methods. Our contributions can be summarized as follows:

- We identify the impacts of visual distractions and develop a controlled, semi-synthetic benchmark DISTRACTEDIT to systematically study this problem in controlled visual scenarios.
- Through extensive evaluation, we reveal that visual distractions cause fragile KE, resulting in severe image memorization where new knowledge fails to generalize.
- We propose SACO, a plug-and-play agentic module that mitigates the effects of distractions. Our experiments demonstrate that visual refinement is a promising direction, and we present SACO as a starting point and hope to motivate more advanced MMKE methods.

2 PRELIMINARIES

Multi-modal Knowledge Editing (MMKE). Let f_θ denote a VLM parameterized by θ. Given an image v and a textual query x, it predicts factual knowledge y, $y = f_\theta(v, x)$. When a specific fact becomes outdated or incorrect, MMKE seeks to update the underlying knowledge triplet (v, x, y) to a new target y^*. The goal is to update the parametric knowledge efficiently so that the edited model produces the updated knowledge, $y^* = f_{\theta'}(v, x)$, without affecting other unrelated knowledge. The editing dataset can be represented as $D_{edit} = \{(v_i, x_i, y_i^*)\}_{i=1}^N$.

Evaluation Protocol. Following established benchmarks (Yao et al., 2023; Cheng et al., 2023; Huang et al., 2024), we comprehensively evaluate editing performance across three core dimensions: Reliability, Generality, and Locality. **Reliability (Rel.)** measures the success of the edit. **Generality** evaluates whether the updated knowledge can successfully transfer to equivalent contexts. The **Image Generality (I-Gen.)** measures how well the edited VLM responds to unseen images containing the same subject. The **Text Generality (T-Gen.)** assesses the model's ability to respond with the new knowledge using rephrased textual prompts. Locality evaluates the preservation of unrelated knowledge. The **Image Locality (I-Loc.)** measures whether the original knowledge is preserved when the edited prompt x is applied to an unedited, unseen image from the same category. The **Text Locality (T-Loc.)** measures whether original knowledge is preserved on the same KE image v when using unrelated textual queries. Calculation details are provided in App. A.1.

MMKE under Visual Distractions. For each subject, we construct an image variant under each visual scenario. The image variants preserve the visual details of the same subject while differing mainly in their visual distractions. For each image variant, we update the same knowledge triplet to the same VLM using the same editing method, and evaluate the edited model under the same protocol. Crucially, the I-Gen. of all variants is measured on the same image of the subject, so that any degradation across scenarios can be attributed to the visual distractions. Real-world images do not reliably provide such controlled image variants. In the following section, we introduce our semi-synthetic approach that constructs these variants with controlled distractions.

3 BENCHMARKING MULTI-MODAL KNOWLEDGE EDITING UNDER VISUAL DISTRACTIONS

This section presents DISTRACTEDIT, which studies how visual distractions affect MMKE. We first describe the curation pipeline (Sec. 3.1) and present our findings (Sec. 3.2).

3.1 DISTRACTEDIT CURATION

DISTRACTEDIT is constructed through a three-stage pipeline, progressively adding visual distractions. An overview of the curation process is illustrated in Fig. 1. We initialize DISTRACTEDIT by defining

174 knowledge seeds. Fig. 2 demonstrates the diversity of the knowledge seeds in our benchmark. More detailed statistics are included in App. A. The pipeline involves curating the three visual scenarios, knowledge triplets, and evaluation set.

Isolated Entity (Original) Curation. At this foundational stage, we conduct a large-scale web crawl to collect candidate images and strictly isolate the desired knowledge by requiring the subject in the knowledge seed to be the dominant foreground element with a clean background. Following rigorous human filtering, we assign two high-quality images for each knowledge seed: a source image $v_{\text{iso}}^{\text{src}}$ used for the KE phase, and a hold-out test image $v_{\text{iso}}^{\text{test}}$ reserved for evaluating image generality. Notably, $v_{\text{iso}}^{\text{test}}$ is deliberately kept clean so that any degradation can be attributed to the visual distractions rather than other factors such as the difficulty of recognizing the subject in a visually distracted image.

Contextual Background (Complex Scene) Curation. We synthesize Contextual Background images through an automated, VLM-guided pipeline. Taking the clean image $v_{\text{iso}}^{\text{src}}$ and the associated subject in x as input, we prompt Gemini 3 series (Google DeepMind, 2026) to design a highly plausible, real-world scene tailored to the subject. Leveraging Gemini's own world knowledge, the model generates a detailed scene description, which specifies contextually appropriate backgrounds and environmental elements. To preserve the original details in $v_{\text{iso}}^{\text{src}}$, the description and $v_{\text{iso}}^{\text{src}}$ are then passed as an instruction to Nano Banana to synthesize the contextual image v_{ctx}.

Multi-Subject Interference (Multi-Subj) Curation. We further simulate scenes in which multiple semantically related subjects appear simultaneously. To reflect the complexity in the real-world images, we take the complex scene image v_{ctx} as the visual input for a similar VLM-guided framework. We further prompt Gemini to analyze the existing scene and design a scene that introduces visually similar or semantically related subjects alongside the original subject.

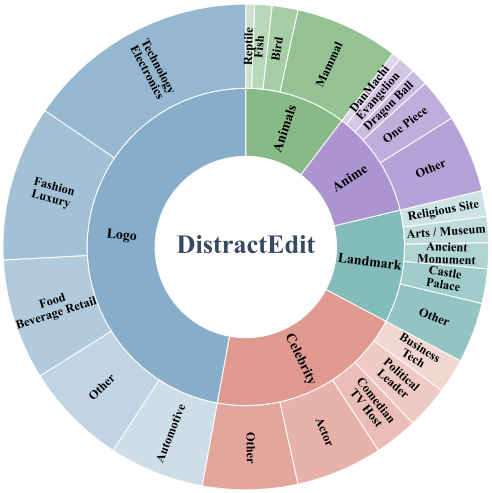

Figure 2: Statistics of DISTRACTEDIT.

Knowledge Triplet Curation. Since pre-trained VLMs encode varying factual knowledge, directly using real-world facts for KE is risky. If a model already knows the target fact, the editing process becomes entirely redundant, making all subsequent evaluations invalid. To strictly prevent this and ensure a controlled setup, we prompt Gemini 3 to generate plausible but counterfactual updates as KE targets. For example, given an image of a *hippopotamus* with the query *"What animal is this?"*, the updated knowledge is *"Rhinoceros"*, a semantically related yet counterfactual answer that no model would produce by default.

Finally, three human annotators conduct comprehensive manual quality checks, verifying images and knowledge triples to ensure that they are logically sound and plausible. We include more details and qualitative demonstrations by visually comparing with real-world images in App. A.

3.2 BENCHMARK RESULTS ON DISTRACTEDIT

Experiment Setup. To systematically evaluate MMKE performance, we benchmark eight editing methods: 1) FT-L (Meng et al., 2022a), 2) IKE (Zheng et al., 2023), 3) MEND (Mitchell et al., 2021), 4) LoRA (Hu et al., 2021), 5) WISE (Wang et al., 2024b), 6) GRACE (Hartvigsen et al., 2023), 7) BalancEdit (Guo et al., 2025), and 8) VisEdit (Chen et al., 2025b). Our experiments are conducted using EasyEdit (Wang et al., 2024c) or their official codebases. We select two VLM backbones: 1) InternVL-3.5-8B (InternVL, Wang et al. (2025)), reflecting the current state of VLM advancements. 2) Blip2-OPT 2.7B (Blip2, Li et al. (2023)), chosen for its popularity in the MMKE community.

Following Cheng et al. (2023); Huang et al. (2024), we evaluate the editing performance in terms of editing success, generality, and locality based on the definition detailed in Sec. 2. More implementation details can be found in App. A.3. Tab. 1 presents the editing performance of all methods under the three visual scenarios. We organize our analysis around four key findings.

Method	Metric	InternVL-3.5 8B			Blip2-OPT 2.7B		
		Original	Complex Scene	Multi-Subj	Original	Complex Scene	Multi-Subj
FT-L	Rel.	1.00	$1.00_{+0.0\%}$	$1.00_{+0.0\%}$	0.88	$0.93_{+5.7\%}$	$0.94_{+6.8\%}$
	T-Gen.	0.99	$1.00_{+1.0\%}$	$1.00_{+1.0\%}$	0.56	$0.61_{+8.9\%}$	$0.70_{+25.0\%}$
	I-Gen.	0.99	$0.99_{+0.0\%}$	$0.93_{-6.1\%}$	0.69	$0.51_{-26.1\%}$	$0.45_{-34.8\%}$
	T-Loc.	0.54	$0.51_{-5.6\%}$	$0.42_{-22.2\%}$	0.82	$0.83_{+1.2\%}$	$0.76_{-7.3\%}$
	I-Loc.	0.52	$0.52_{+0.0\%}$	$0.54_{+3.8\%}$	0.86	$0.91_{+5.8\%}$	$0.87_{+1.2\%}$
	Avg.	0.81	$0.80_{-0.5\%}$	$0.78_{-3.7\%}$	0.76	$0.76_{-0.5\%}$	$0.74_{-2.4\%}$
IKE	Rel.	0.49	$0.49_{+0.0\%}$	$0.48_{-2.0\%}$	0.98	$1.00_{+2.0\%}$	$0.99_{+1.0\%}$
	T-Gen.	0.49	$0.48_{-2.0\%}$	$0.49_{+0.0\%}$	0.40	$0.43_{+7.5\%}$	$0.45_{+12.5\%}$
	I-Gen.	0.49	$0.49_{+0.0\%}$	$0.49_{+0.0\%}$	0.98	$1.00_{+2.0\%}$	$0.99_{+1.0\%}$
	T-Loc.	0.91	$0.88_{-3.3\%}$	$0.86_{-5.5\%}$	0.96	$0.82_{-14.6\%}$	$0.84_{-12.5\%}$
	I-Loc.	0.73	$0.72_{-1.4\%}$	$0.70_{-4.1\%}$	0.27	$0.26_{-3.7\%}$	$0.20_{-25.9\%}$
	Avg.	0.62	$0.61_{-1.6\%}$	$0.60_{-2.9\%}$	0.72	$0.70_{-2.2\%}$	$0.69_{-3.3\%}$
MEND	Rel.	0.91	$0.83_{-8.8\%}$	$0.81_{-11.0\%}$	0.98	$0.98_{+0.0\%}$	$0.98_{+0.0\%}$
	T-Gen.	0.87	$0.83_{-4.6\%}$	$0.84_{-3.4\%}$	0.94	$0.94_{+0.0\%}$	$0.95_{+1.0\%}$
	I-Gen.	0.89	$0.79_{-11.2\%}$	$0.72_{-19.1\%}$	0.98	$0.95_{-3.0\%}$	$0.93_{-5.0\%}$
	T-Loc.	0.90	$0.88_{-2.2\%}$	$0.87_{-3.3\%}$	0.97	$0.95_{-2.0\%}$	$0.92_{-5.0\%}$
	I-Loc.	0.74	$0.74_{+0.0\%}$	$0.77_{+4.1\%}$	0.76	$0.76_{+0.0\%}$	$0.74_{-2.0\%}$
	Avg.	0.86	$0.81_{-5.6\%}$	$0.80_{-7.0\%}$	0.93	$0.92_{-1.1\%}$	$0.90_{-2.4\%}$
LoRA	Rel.	0.99	$1.00_{+1.0\%}$	$1.00_{+1.0\%}$	0.97	$0.98_{+1.0\%}$	$0.98_{+1.0\%}$
	T-Gen.	0.65	$0.72_{+10.8\%}$	$0.74_{+13.8\%}$	0.95	$0.94_{-1.1\%}$	$0.95_{+0.0\%}$
	I-Gen.	0.84	$0.63_{-25.0\%}$	$0.51_{-39.3\%}$	0.94	$0.94_{+0.0\%}$	$0.90_{-4.3\%}$
	T-Loc.	1.00	$0.99_{-1.0\%}$	$0.97_{-3.0\%}$	0.30	$0.35_{+16.7\%}$	$0.27_{-10.0\%}$
	I-Loc.	0.86	$0.94_{+9.3\%}$	$0.96_{+11.6\%}$	0.54	$0.62_{+14.8\%}$	$0.64_{+18.5\%}$
	Avg.	0.87	$0.86_{-1.4\%}$	$0.84_{-3.7\%}$	0.74	$0.77_{+3.5\%}$	$0.75_{+1.1\%}$
WISE	Rel.	1.00	$0.97_{-3.0\%}$	$0.99_{-1.0\%}$	0.91	$0.91_{+0.0\%}$	$0.87_{-4.4\%}$
	T-Gen.	0.84	$0.84_{+0.0\%}$	$0.83_{-1.2\%}$	0.52	$0.46_{-11.5\%}$	$0.53_{+1.9\%}$
	I-Gen.	0.97	$0.71_{-26.8\%}$	$0.47_{-51.5\%}$	0.82	$0.56_{-31.7\%}$	$0.49_{-40.2\%}$
	T-Loc.	0.98	$0.98_{+0.0\%}$	$0.98_{+0.0\%}$	0.46	$0.48_{+4.3\%}$	$0.47_{+2.2\%}$
	I-Loc.	0.71	$0.93_{+31.0\%}$	$0.96_{+35.2\%}$	0.32	$0.37_{+15.6\%}$	$0.34_{+6.3\%}$
	Avg.	0.90	$0.89_{-1.6\%}$	$0.85_{-6.0\%}$	0.61	$0.56_{-8.3\%}$	$0.54_{-10.9\%}$
GRACE	Rel.	1.00	$1.00_{+0.0\%}$	$0.99_{-1.0\%}$	0.93	$0.91_{-2.2\%}$	$0.90_{-3.2\%}$
	T-Gen.	0.77	$0.77_{+0.0\%}$	$0.78_{+1.3\%}$	0.65	$0.58_{-10.8\%}$	$0.57_{-12.3\%}$
	I-Gen.	0.65	$0.53_{-18.5\%}$	$0.46_{-29.2\%}$	0.89	$0.79_{-11.2\%}$	$0.71_{-20.2\%}$
	T-Loc.	0.93	$0.93_{+0.0\%}$	$0.91_{-2.2\%}$	0.96	$0.95_{-1.0\%}$	$0.97_{+1.0\%}$
	I-Loc.	0.78	$0.83_{+6.4\%}$	$0.89_{+14.1\%}$	0.72	$0.87_{+20.8\%}$	$0.88_{+22.2\%}$
	Avg.	0.83	$0.81_{-1.7\%}$	$0.81_{-2.4\%}$	0.83	$0.82_{-1.2\%}$	$0.81_{-2.9\%}$
BalancEdit	Rel.	1.00	$1.00_{+0.0\%}$	$1.00_{+0.0\%}$	0.87	$0.94_{+8.0\%}$	$0.94_{+8.0\%}$
	T-Gen.	0.90	$0.98_{+8.9\%}$	$0.97_{+7.8\%}$	0.56	$0.61_{+8.9\%}$	$0.71_{+26.8\%}$
	I-Gen.	0.97	$0.82_{-15.5\%}$	$0.63_{-35.1\%}$	0.63	$0.31_{-50.8\%}$	$0.29_{-54.0\%}$
	T-Loc.	0.98	$0.97_{-1.0\%}$	$0.94_{-4.1\%}$	0.82	$0.81_{-1.2\%}$	$0.76_{-7.3\%}$
	I-Loc.	0.65	$0.84_{+29.2\%}$	$0.91_{+40.0\%}$	0.97	$0.97_{+0.0\%}$	$0.97_{+0.0\%}$
	Avg.	0.90	$0.92_{+2.4\%}$	$0.89_{-1.1\%}$	0.77	$0.73_{-5.5\%}$	$0.73_{-4.7\%}$
VisEdit	Rel.	–	–	–	0.89	$0.89_{+0.0\%}$	$0.89_{+0.0\%}$
	T-Gen.	–	–	–	0.86	$0.85_{-1.2\%}$	$0.87_{+1.2\%}$
	I-Gen.	–	–	–	0.88	$0.89_{+1.1\%}$	$0.89_{+1.1\%}$
	T-Loc.	–	–	–	0.94	$0.81_{-13.8\%}$	$0.85_{-9.6\%}$
	I-Loc.	–	–	–	0.24	$0.23_{-4.2\%}$	$0.23_{-4.2\%}$
	Avg.	–	–	–	0.76	$0.73_{-3.7\%}$	$0.75_{-2.1\%}$

Table 1: Benchmark results of editing performance. Degraded scores are highlighted in red.

Finding 1: Visual distraction broadly undermines editing performance. Across most editing methods and both VLM backbones, transitioning from the clean *Original* visual condition to the *Complex Scene* and *Multi-Subj* settings leads to a general decline in overall editing performance. On InternVL, the average performance drops by up to 7.0% (MEND), and on Blip2, the degradation reaches 10.9% (WISE). This trend widely holds across almost all editing paradigms, including meta-learning (MEND), weight-routing (WISE), and memory-based (GRACE) approaches. This observation establishes that visual distraction is a universal bottleneck for MMKE.

Finding 2: Visual distraction disproportionately disrupts knowledge association. A closer inspection reveals a striking asymmetry between text-side and image-side metrics. In the majority of methods, **T-Gen.** and **T-Loc.** remain largely stable under degraded visual conditions. In contrast, Image Generality (**I-Gen.**) suffers severe collapses in most cases. For instance, WISE drops from 0.97 to 0.47 on InternVL, and BalancEdit on Blip2 drops from 0.63 to 0.29. Crucially, these **I-Gen.** declines are often accompanied by a simultaneous rise in I-Loc. (e.g., WISE and BalancEdit on InternVL increase by around 40%). IKE, which performs knowledge updates through in-context learning, demonstrates a different failure pattern. Although **I-Gen.** remains stable, **I-Loc.** and **T-Loc.** suffer declines, indicating visual distractions cause IKE to overwrite the original unrelated knowledge.

These patterns collectively reveal that ***visual distraction causes a shallow and fragile editing process, leading to a blind memorization of a surface-level image pattern***. The edited knowledge is weakly encoded, which often is only triggered under the exact visual conditions.

Finding 3: The impact of visual distraction is mechanism- and backbone-specific. While the overall trend favoring cleaner visuals is consistent, the magnitude and pattern of degradation vary substantially across editing methods and model backbones. The optimal visual condition is not uniform: while most methods perform best under *Original*, certain method-backbone combinations show comparable or even slightly better results under *Complex Scene*. For instance, LoRA applied to Blip2 demonstrates robustness, suffering no performance degradation. Such complex dependencies suggest the need for future studies on how VLMs' internal knowledge manifests under diverse visual conditions and how these mechanisms can inform more principled and robust MMKE design.

Finding 4: Different types of visual distraction induce distinct failure modes. Comparing the two degraded conditions, *Multi-Subj* generally inflicts more severe damage than *Complex Scene*, particularly on **I-Gen.** This disparity suggests the two conditions disrupt editing through different mechanisms: *Complex Scene* introduces background clutter that dilutes the visual signal, while *Multi-Subj* creates subject-level ambiguity that confuses which entity the new knowledge should bind to. The latter poses more fundamental challenges to knowledge grounding, as the editing process must extract relevant visual cues and resolve *which* visual entity is the target of the knowledge update.

To further support our findings, we conduct a consistency study using small-scale real-world images. The results show trends consistent with those on DISTRACTEDIT. We provide the results in App. A.5.

4 SACO: AN AGENTIC IMAGE REFINEMENT APPROACH

Driven by findings from Sec. 3, we propose SACO, an agentic solution to See, Analyze, Clean, and Optimize visuals to enhance visual quality. As shown in Fig. 3, SACO involves two agents to identify the knowledge and modify the visual input to ensure that the knowledge can be better internalized by the edited models.

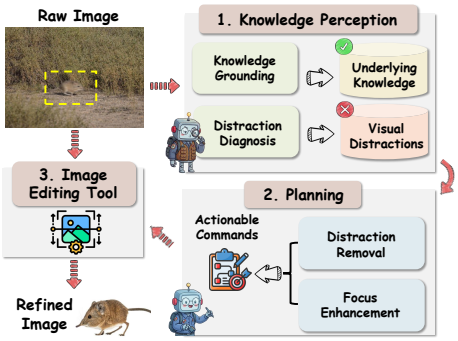

Figure 3: The proposed SACO overview.

Knowledge Perception Agent ($\mathcal{A}_{per}$) takes a raw image v and an editing request x to generate a structured diagnostic analysis $\mathcal{C} = \mathcal{A}_{per}(v, x)$ on visual cues corresponding to the knowledge and visual cues that are potential distractions. This entails two sub-goals: First, **knowledge grounding**: the agent analyzes the image to identify the most essential visual cues that correspond to the knowledge by leveraging its own world knowledge. Second, **visual distraction diagnosis**: the agent inspects the visual input to analyze and pinpoint specific visual elements that could cause the model to memorize specific visual cues.

Planner Agent ($\mathcal{A}_{plan}$) takes the image v, the editing request x, and the report $\mathcal{C}$ to formulate an actionable plan $\mathcal{I} = \mathcal{A}_{plan}(v, x, \mathcal{C})$. This produces a *visual scene* design through two sub-tasks: First, **distraction removal**: the agent specifies which distracting elements to eliminate and determines what contextually appropriate content should replace them. Second, **focus enhancement**: the agent preserves and strengthens the visual cues corresponding to the target knowledge.

Image Editing Tool ($\mathcal{T}_{edit}$) executes the scene design plan $\mathcal{I}$ on the raw image v, thereby producing a refined image $v^* = \mathcal{T}_{edit}(v, \mathcal{I})$. We employ Nano Banana, which modifies only the elements specified in the plan. The resulting v^* replaces the image as the input to the editing method.

SACO provides a plug-and-play solution that can be utilized by different editing methods. More details are in App. B.3.

5 PERFORMANCE OF SACO

We evaluate SACO on four KE methods: MEND, WISE, GRACE, and BalancEdit, which exhibited notable sensitivity to visual distractions.

5.1 QUANTITATIVE RESULTS

Tab. 2 presents the quantitative results of incorporating SACO to enhance KE performance under Multi-Subj. *Raw visual* columns denote the performance under the visual distraction settings. For reference, we further report the corresponding *BoA* (Best-of-All) results across the three visual scenarios (derived from Tab. 1), which serve as an empirical KE performance upper bound. We include the performance of SACO under more visual scenarios in App. B.1.

Method	Metric	InternVL-3.5 8B			Blip2-OPT 2.7B		
		BoA	**Raw Visual**	**Refined Visual**	***BoA***	**Raw Visual**	**Refined Visual**
MEND	**Rel.**	*0.91*	0.81	0.87	*0.98*	0.98	0.98
	T-Gen.	*0.87*	0.84	0.87	*0.95*	0.95	0.95
	I-Gen.	*0.89*	0.72	0.81	*0.98*	0.93	0.97
	T-Loc.	*0.90*	0.87	0.87	*0.97*	0.92	0.95
	I-Loc.	*0.77*	0.77	0.77	*0.76*	0.74	0.73
	Avg	*0.87*	0.80	0.84	*0.93*	0.90	0.92
WISE	**Rel.**	*1.00*	0.99	1.00	*0.91*	0.87	0.90
	T-Gen.	*0.84*	0.83	0.86	*0.53*	0.53	0.49
	I-Gen.	*0.97*	0.47	0.90	*0.82*	0.49	0.74
	T-Loc.	*0.98*	0.98	0.95	*0.48*	0.47	0.44
	I-Loc.	*0.96*	0.96	0.71	*0.37*	0.34	0.31
	Avg	*0.95*	0.85	0.88	*0.62*	0.54	0.58
GRACE	**Rel.**	*1.00*	0.99	1.00	*0.93*	0.90	0.93
	T-Gen.	*0.78*	0.78	0.78	*0.65*	0.57	0.61
	I-Gen.	*0.65*	0.46	0.60	*0.89*	0.71	0.89
	T-Loc.	*0.93*	0.91	0.92	*0.97*	0.97	0.94
	I-Loc.	*0.89*	0.89	0.89	*0.88*	0.88	0.71
	Avg	*0.85*	0.81	0.84	*0.86*	0.81	0.82
BalancEdit	**Rel.**	*1.00*	1.00	1.00	*0.94*	0.94	1.00
	T-Gen.	*0.98*	0.97	0.96	*0.71*	0.71	0.55
	I-Gen.	*0.97*	0.63	0.93	*0.63*	0.29	0.57
	T-Loc.	*0.98*	0.94	0.95	*0.82*	0.76	0.83
	I-Loc.	*0.91*	0.91	0.65	*0.97*	0.97	0.96
	Avg	*0.97*	0.89	0.90	*0.81*	0.73	0.78

Table 2: Performance enhancement from SACO visual refinement on different editing methods under Multi-subj. ***BoA*** reports the *best-of-all* performance in different visual scenarios for reference.

Overall Enhancement. The integration of SACO yields consistent improvements, elevating the average KE performance (Avg) across all evaluated methods and both VLM backbones. For instance, on InternVL, the average KE performance is increased by an absolute margin close to 0.03, translating to a relative improvement of up to 5%. The performance gain on Blip2 is even more prominent, reaching up to around 7%. This consistent upward trend demonstrates the generalizability and plug-and-play capability of SACO across different editing paradigms.

Aspect-Specific Analysis. We next examine the specific aspects of SACO's effect on MMKE methods. As Tab. 2 shows, the performance gains are concentrated in **I-Gen.**. We view this as direct evidence that refining visual inputs helps KE methods establish a stronger visual-knowledge association. This result also aligns with our observations from Sec. 3. While SACO improves the **I-Gen.**, certain method-backbone combinations show tradeoffs between **I-Gen.** and **T-Loc.**, **I-Loc.**, or **T-Gen.**. This suggests that visual refinement alone does not guarantee precise control over the editing or modality scope of the VLMs' knowledge and reaffirms that the *impact of visual distraction is highly mechanism- and backbone-specific*. SACO therefore provides a first step as the mitigation strategy and opens up several future directions, including more advanced agentic structures, adaptive visual refinement, and editing methods that are natively robust to distractions. In short, these results confirm the significance of DISTRACTEDIT, establishing it as a foundational testbed for future research.

5.2 QUALITATIVE RESULTS

We further show qualitative results to demonstrate how SACO addresses visual distractions. Fig. 4 illustrates the pipeline of SACO on a randomly selected example. Given the editing request, SACO first identifies the baboon-related knowledge, exemplified by its characteristics. Meanwhile, the terrain and lighting are categorized as visual distractions. Next, the *planning* phase generates an

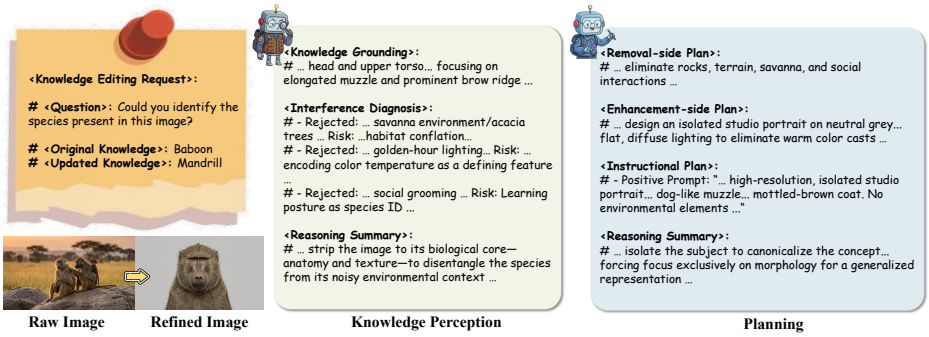

Figure 4: Illustration of the SACO workflow for visual enhancement.

enhancement strategy by removing the identified distractions and refining the subject into an isolated studio portrait with a neutral background and diffuse lighting. The resulting refined image preserves the anatomy and texture critical for knowledge grounding, providing editing methods with a more focused visual input. More visual examples are shown in Fig. 5. By analyzing the editing requests and raw visuals, SACO successfully identifies and enhances edit-relevant subjects across diverse scenarios, removing visual distractions such as backgrounds, lighting conditions, and occluding foregrounds.

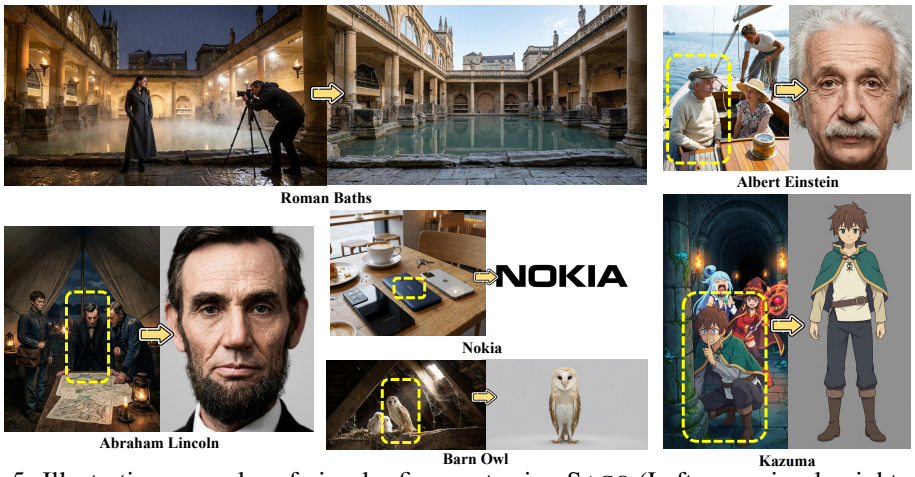

Figure 5: Illustrative examples of visual refinement using SACO (Left: raw visuals, right: refined visuals). Target knowledge concepts and corresponding visual objects are marked.

5.3 ABLATION STUDY

We end this section with an ablation study by presenting variants of SACO. SACO-FE only uses half of the analysis in SACO by using a single VLM agent to solely enforce the knowledge grounding and focus enhancement. SACO-FE generates image-editing instructions to make the target entity more visually salient. SACO-DR uses only the other half of the analysis in SACO. SACO-DR solely enforces the analysis and the removal of visual distractions by generating instructions to the image editor to remove visual distractions. SACO-IC demonstrates the effect of image editing tools by replacing the image editing model with an image cropping tool. We report the variants on InternVL under *Multi-Subj*.

According to Tab. 3, we highlight the following findings. First, using either focus enhancement or distraction removal alone leads to weaker overall editing performance (Avg) than full SACO. Applied alone, SACO-FE re-renders the subject while the distractions still remain, and SACO-DR removes the distractions but the subject remains small.

Method	Metric	SACO-FE	SACO-DR	SACO-IC	SACO
MEND	Rel.	0.79	0.79	0.86	0.87
	T-Gen.	0.72	0.75	0.86	0.87
	I-Gen.	0.73	0.72	0.80	0.81
	T-Loc.	0.88	0.86	0.85	0.87
	I-Loc.	0.64	0.66	0.75	0.77
	Avg.	0.75	0.76	0.82	0.84
WISE	Rel.	0.90	0.98	1.00	1.00
	T-Gen.	0.58	0.74	0.87	0.86
	I-Gen.	0.15	0.53	0.91	0.90
	T-Loc.	0.81	0.94	0.96	0.95
	I-Loc.	0.91	0.61	0.72	0.71
	Avg.	0.66	0.76	0.89	0.88
GRACE	Rel.	1.00	0.99	1.00	1.00
	T-Gen.	0.68	0.67	0.79	0.78
	I-Gen.	0.18	0.05	0.63	0.60
	T-Loc.	0.79	0.90	0.92	0.92
	I-Loc.	0.79	0.93	0.88	0.89
	Avg.	0.69	0.71	0.84	0.84
BalancEdit	Rel.	1.00	1.00	0.99	1.00
	T-Gen.	0.95	0.97	0.54	0.96
	I-Gen.	0.46	0.66	0.60	0.93
	T-Loc.	0.76	0.91	0.82	0.95
	I-Loc.	0.75	0.43	0.96	0.65
	Avg.	0.79	0.79	0.78	0.90

Table 3: Ablation study of SACO.

This demonstrates their combination in SACO is necessary to improve current MMKE methods. Additionally, for most methods, SACO-IC is nearly sufficient because cropping retains subject details and a dominant view. However, cropping is clearly insufficient for BalancEdit. This further supports that the impact of visual distraction is mechanism-specific. Additionally, since cropping is more efficient, as mentioned earlier, one future direction is to design more advanced agentic system to select from multiple tools. We include more details, qualitative results, and cost analysis in App. B.2.

6 RELATED WORK

Knowledge Editing for LLMs. Existing methods generally fall into internal and external storage approaches. To internalize new knowledge, the locate-and-edit paradigm (Meng et al., 2022a;b; Li et al., 2024b; Gupta et al., 2024; Fang et al., 2024; Hu et al., 2024; Gu et al., 2024; Ma et al., 2025) identifies target updates for model weights to intervene in specific model behaviors, and other studies have designed parameter-efficient fine-tuning (PEFT) methods (Hu et al., 2021; Zhang et al., 2023; Wu et al., 2024; Zhang et al., 2024c; Wang et al., 2024a; Liu et al., 2025b) that can be trained with specific new knowledge statements directly and precisely. External storage approaches resort to external memories without updating the original parameters, leveraging hypernetworks (Mitchell et al., 2021; Tan et al., 2024), retrieval-augmented generation (Zhang et al., 2024b; Jiang et al., 2024; Wei et al., 2024; Wang et al., 2024e; Chen et al., 2025a), and routing mechanisms on separately learned weights or module copies (De Cao et al., 2021; Mitchell et al., 2022; Hartvigsen et al., 2023; Wang et al., 2024b; Yu et al., 2024). However, they were primarily designed for the textual domain.

Knowledge Editing for VLMs. Recent efforts have extended KE methods to Vision-Language Models (VLMs). Initial research primarily migrated existing text editors directly to the multi-modal space (Cheng et al., 2023; Wang et al., 2024c; Huang et al., 2024; Zhang et al., 2025). Subsequent studies have introduced VLM-specific strategies, such as analyzing the structural influence of visual semantics towards better editing performance (Guo et al., 2025; Chen et al., 2025b; Shi et al., 2025). In addition, to better evaluate these multi-modal capabilities, comprehensive benchmarks such as MMEdit (Cheng et al., 2023), VLKEB (Huang et al., 2024), MIKE (Li et al., 2024a), LiveVQA (Fu et al., 2026), and MC-MKE (Zhang et al., 2025) have been developed. However, they do not systematically study the effects of visual distractions.

Knowledge Editing Overfitting. Knowledge Editing updates a large pre-trained model with one or a few statements conveying the new knowledge. As a result, existing methods often *overfit* to the provided *statements*, compromising the generalizability of the edited model (Zhang et al., 2024a; Ma et al., 2024; Cohen et al., 2024; Qi et al., 2024; Lampinen et al., 2025; Liu et al., 2025a). Recent works have identified various forms of KE overfitting and proposed respective mitigation strategies. To name a few, Zhang et al. (2024a) and Qi et al. (2024) utilize in-context prompted distributions as the fitting target (Lampinen et al., 2025) for better generalizability. Ma et al. (2024) investigate the perturbation of neighboring knowledge caused by answer-level overfitting. Liu et al. (2025a) reveal the mechanistic vulnerabilities of heterogeneous overfitting when editing pre-trained models with small, localized datasets. However, these efforts largely focus on text-only editing. More recently, Li et al. (2026), Das & Jena (2026) and Wang et al. (2026) aim to improve the generalization. However, they are evaluated on standard MMKE benchmarks that do not characterize how visual distractions affect the edit. They are also orthogonal to our input-side refinement SACO, which can be combined with these methods. In this work, we reveal that overfitting is profoundly exacerbated when the new knowledge is presented with suboptimal visual evidence (e.g. with visual distractions).

7 CONCLUSION

We study the challenge of Multi-modal Knowledge Editing (MMKE) on real-world images containing visual distractions. We introduce DISTRACTEDIT, a benchmark designed to systematically evaluate the impact of visual distractions through controlled visual scenarios: Isolated Entity, Contextual Background, and Multi-Subject Interference. Our extensive evaluations demonstrate that visual distractions broadly undermine editing performance and cause existing methods to memorize image-specific artifacts. To address this bottleneck, we propose SACO, a plug-and-play agentic solution that leverages VLM agents to see, analyze, clean, and optimize visual inputs prior to editing. By actively removing irrelevant and distracting contextual subjects, SACO isolates the core visual concept.

AI USE STATEMENT

In this work, we used generative AI tools for polishing writing, searching for related works, discussing ideas, and generating synthetic datasets. We have not used generative AI tools for drafting the paper or providing mathematical claims. Additionally, we used generative AI tools for implementing and debugging some parts of our codebase. We have reviewed all AI-assisted work. For example, we manually validate the generated images from AI, reviewed the code, and verified the polished text. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

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

Qizhou Chen, Taolin Zhang, Xiaofeng He, Dongyang Li, Chengyu Wang, Longtao Huang, and Hui Xue. Lifelong knowledge editing for llms with retrieval-augmented continuous prompt learning, 2025a.

Qizhou Chen, Taolin Zhang, Chengyu Wang, Xiaofeng He, Dakan Wang, and Tingting Liu. Attribution analysis meets model editing: Advancing knowledge correction in vision language models with visedit. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 2168–2176, 2025b.

Siyuan Cheng, Bozhong Tian, Qingbin Liu, Xi Chen, Yongheng Wang, Huajun Chen, and Ningyu Zhang. Can we edit multimodal large language models? In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 13877–13888, 2023.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. Evaluating the ripple effects of knowledge editing in language models. *Transactions of the Association for Computational Linguistics*, 12:283–298, 2024.

Gyanendra Das and Sai Jena. Dsca: Dynamic subspace concept alignment for lifelong vlm editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 40772–40781, 2026.

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

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Yuntao Du, Kailin Jiang, Zhi Gao, Chenrui Shi, Zilong Zheng, Siyuan Qi, and Qing Li. Mmke-bench: A multimodal editing benchmark for diverse visual knowledge. In *The Thirteenth International Conference on Learning Representations*, 2025.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Shi Jie, Xiang Wang, Xiangnan He, and Tat-Seng Chua. Alphaedit: Null-space constrained knowledge editing for language models. *arXiv preprint arXiv:2410.02355*, 2024.

Mingyang Fu, Yuyang Peng, Dongping Chen, Zetong Zhou, Benlin Liu, Yao Wan, Zhou Zhao, Philip S Yu, and Ranjay Krishna. Seeking and updating with live visual knowledge. *Advances in Neural Information Processing Systems*, 38, 2026.

Cristina Garcia. Is the elephant shrew a shrew, an elephant, or neither? https://www.travel4wildlife.com/elephant-shrew-what-are-sengis/, Feb 2021. Last updated: March 24, 2023. Accessed: 2026-03-31.

Google DeepMind. Gemini 3, 2026. URL https://gemini.google.com. Large language model developed by Google.

Jia-Chen Gu, Hao-Xiang Xu, Jun-Yu Ma, Pan Lu, Zhen-Hua Ling, Kai-Wei Chang, and Nanyun Peng. Model editing harms general abilities of large language models: Regularization to the rescue. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 16801–16819, 2024.

Dongliang Guo, Mengxuan Hu, Zihan Guan, Thomas Hartvigsen, and Sheng Li. Balancedit: Dynamically balancing the generality-locality trade-off in multi-modal model editing. *arXiv preprint arXiv:2505.01343*, 2025.

Akshat Gupta, Dev Sajnani, and Gopala Anumanchipalli. A unified framework for model editing. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 15403–15418, 2024.

Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. Aging with grace: Lifelong model editing with discrete key-value adaptors. *Advances in neural information processing systems*, 36:47934–47959, 2023.

Chenhui Hu, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. Wilke: Wise-layer knowledge editor for lifelong knowledge editing, 2024.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

Han Huang, Haitian Zhong, Tao Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. Vlkeb: A large vision-language model knowledge editing benchmark. *Advances in Neural Information Processing Systems*, 37:9257–9280, 2024.

Yuxin Jiang, Yufei Wang, Chuhan Wu, Wanjun Zhong, Xingshan Zeng, Jiahui Gao, Liangyou Li, Xin Jiang, Lifeng Shang, Ruiming Tang, et al. Learning to edit: Aligning llms with knowledge editing. *arXiv preprint arXiv:2402.11905*, 2024.

Andrew K Lampinen, Arslan Chaudhry, Stephanie CY Chan, Cody Wild, Diane Wan, Alex Ku, Jörg Bornschein, Razvan Pascanu, Murray Shanahan, and James L McClelland. On the generalization of language models from in-context learning and finetuning: a controlled study. *arXiv preprint arXiv:2505.00661*, 2025.

Jiaqi Li, Miaozeng Du, Chuanyi Zhang, Yongrui Chen, Nan Hu, Guilin Qi, Haiyun Jiang, Siyuan Cheng, and Bozhong Tian. Mike: A new benchmark for fine-grained multimodal entity knowledge editing. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 5018–5029, 2024a.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023.

Siyuan Li, Youyuan Zhang, Ruitong Liu, Junxi Wang, and Jing Li. Multimodal knowledge edit-scoped generalization for online recursive mllm editing. *arXiv preprint arXiv:2607.01978*, 2026.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. Pmet: Precise model editing in a transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 18564–18572, 2024b.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Tianci Liu, Ruirui Li, Zihan Dong, Hui Liu, Xianfeng Tang, Qingyu Yin, Linjun Zhang, Haoyu Wang, and Jing Gao. Mitigating heterogeneous token overfitting in llm knowledge editing. *arXiv preprint arXiv:2502.00602*, 2025a.

Tianci Liu, Ruirui Li, Haoyu Wang, Yunzhe Qi, Hui Liu, Xianfeng Tang, Tianqi Zheng, Qingyu Yin, Monica Xiao Cheng, Jun Huan, and Jing Gao. Unlocking efficient, scalable, and continual knowledge editing with basis-level representation fine-tuning. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Jun-Yu Ma, Zhen-Hua Ling, Ningyu Zhang, and Jia-Chen Gu. Neighboring perturbations of knowledge editing on large language models. *arXiv preprint arXiv:2401.17623*, 2024.

Jun-Yu Ma, Hong Wang, Hao-Xiang Xu, Zhen-Hua Ling, and Jia-Chen Gu. Perturbation-restrained sequential model editing, 2025.

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

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

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

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

Siyuan Qi, Bangcheng Yang, Kailin Jiang, Xiaobo Wang, Jiaqi Li, Yifan Zhong, Yaodong Yang, and Zilong Zheng. In-context editing: Learning knowledge from self-induced distributions. *arXiv preprint arXiv:2406.11194*, 2024.

Zhiyi Shi, Binjie Wang, Chongjie Si, Yichen Wu, Junsik Kim, and Hanspeter Pfister. Dualedit: Dual editing for knowledge updating in vision-language models. *arXiv preprint arXiv:2506.13638*, 2025.

Chenmien Tan, Ge Zhang, and Jie Fu. Massive editing for large language models via meta learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Haoyu Wang, Tianci Liu, Ruirui Li, Monica Cheng, Tuo Zhao, and Jing Gao. Roselora: Row and column-wise sparse low-rank adaptation of pre-trained language model for knowledge editing and fine-tuning. *arXiv preprint arXiv:2406.10777*, 2024a.

Haoyuan Wang, Xiaohao Liu, Jiajie Su, Jianmao Xiao, and Chaochao Chen. Beyond binary edits robust multimodal knowledge editing with adversarial subspace alignment. *arXiv preprint arXiv:2605.23780*, 2026.

Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. Wise: Rethinking the knowledge memory for lifelong model editing of large language models. *arXiv preprint arXiv:2405.14768*, 2024b.

Peng Wang, Ningyu Zhang, Bozhong Tian, Zekun Xi, Yunzhi Yao, Ziwen Xu, Mengru Wang, Shengyu Mao, Xiaohan Wang, Siyuan Cheng, Kangwei Liu, Yuansheng Ni, Guozhou Zheng, and Huajun Chen. Easyedit: An easy-to-use knowledge editing framework for large language models, 2024c. URL https://arxiv.org/abs/2308.07269.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. Knowledge editing for large language models: A survey. *ACM Computing Surveys*, 57(3):1–37, 2024d.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

Yiwei Wang, Muhao Chen, Nanyun Peng, and Kai-Wei Chang. Deepedit: Knowledge editing as decoding with constraints. *arXiv preprint arXiv:2401.10471*, 2024e.

Zihao Wei, Liang Pang, Hanxing Ding, Jingcheng Deng, Huawei Shen, and Xueqi Cheng. Stable knowledge editing in large language models. *arXiv preprint arXiv:2402.13048*, 2024.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Reft: Representation finetuning for language models. *arXiv preprint arXiv:2404.03592*, 2024.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. Editing large language models: Problems, methods, and opportunities. *arXiv preprint arXiv:2305.13172*, 2023.

Lang Yu, Qin Chen, Jie Zhou, and Liang He. Melo: Enhancing model editing with neuron-indexed dynamic lora. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19449–19457, 2024.

Zhen Zeng, Leijiang Gu, Xun Yang, Zhangling Duan, Zenglin Shi, and Meng Wang. Visual-oriented fine-grained knowledge editing for multimodal large language models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2491–2500. IEEE, 2025.

Junzhe Zhang, Huixuan Zhang, Xunjian Yin, Baizhou Huang, Xu Zhang, Xinyu Hu, and Xiaojun Wan. Mc-mke: A fine-grained multimodal knowledge editing benchmark emphasizing modality consistency. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 17430–17445, 2025.

Mengqi Zhang, Xiaotian Ye, Qiang Liu, Pengjie Ren, Shu Wu, and Zhumin Chen. Uncovering overfitting in large language model editing. *arXiv preprint arXiv:2410.07819*, 2024a.

Ningyu Zhang, Bozhong Tian, Siyuan Cheng, Xiaozhuan Liang, Yi Hu, Kouying Xue, Yanjie Gou, Xi Chen, and Huajun Chen. Instructedit: Instruction-based knowledge editing for large language models. *arXiv preprint arXiv:2402.16123*, 2024b.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, et al. A comprehensive study of knowledge editing for large language models. *arXiv preprint arXiv:2401.01286*, 2024c.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*, 2023.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2):1–124, 2023.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. Can we edit factual knowledge by in-context learning? *arXiv preprint arXiv:2305.12740*, 2023.

APPENDIX

A DETAILS ON BENCHMARKING MULTI-MODAL KNOWLEDGE EDITING UNDER VISUAL DISTRACTIONS

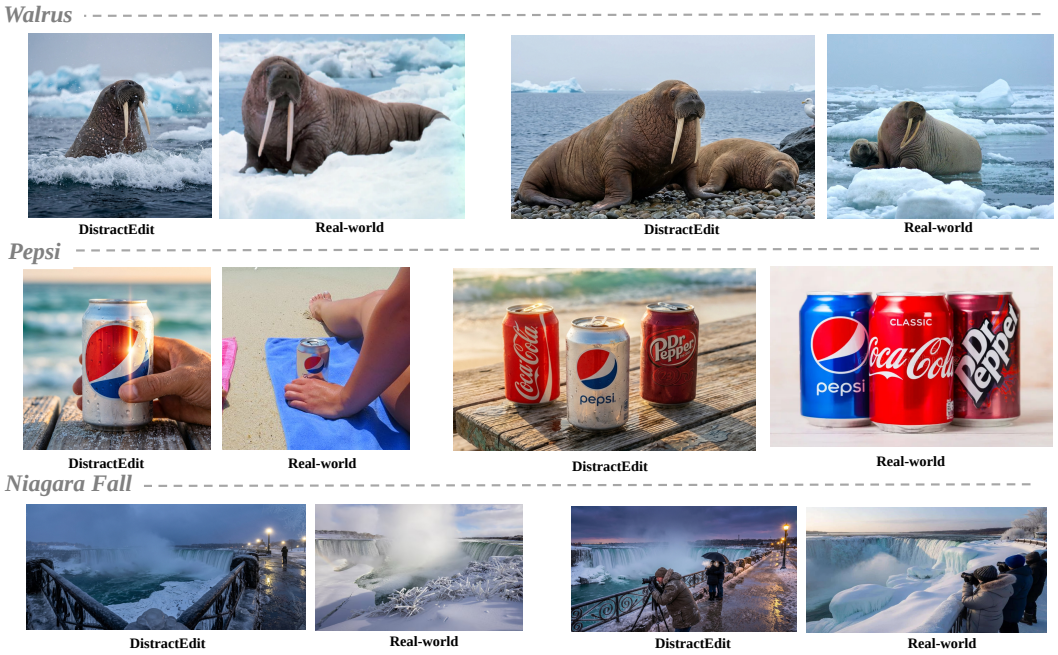

Figure 6: Examples of images from DISTRACTEDIT alongside real-world images.

We collect 174 distinct knowledge seeds and construct DISTRACTEDIT. Detailed statistics are given in Tab. 4. For each seed, we collect real-world images through web crawling. Human annotators manually select and inspect these images to verify that they satisfy the *Isolated Entity (Original)* setting, in which the target subject is clearly visible, dominates the foreground, and appears in a clean background with minimal distracting visual elements. Starting from each source image, we apply our semi-synthetic curation framework. Gemini 3 generates knowledge triplets and scene-specific editing instructions, which are executed by Nano Banana. This process produces visual variants of the same subject, providing controlled comparisons across different distraction scenarios in DISTRACTEDIT. Three human annotators review the resulting images and the knowledge triplets for logical consistency and real-world plausibility. We include the annotation criteria in A.4.

Category	Knowledge Seeds	Percentage
Logo	82	47.1%
Celebrity	35	20.1%
Landmark	20	11.5%
Anime	19	10.9%
Animals	18	10.3%
Total	**174**	**100.0%**

Table 4: Statistics of knowledge seeds.

We additionally collect real-world images for qualitative comparison with the curated images. As shown in Fig. 6, the side-by-side examples illustrate that the semi-synthesized images are visually plausible and close to real-world images. They also demonstrate that our proposed visual scenarios indeed exist in real-world images.

In addition to the knowledge-editing triplets, we curate Text-locality (T-Loc.) questions about knowledge unrelated to the knowledge editing request within the same image. For example, a landmark editing request may ask "What building is this?", whereas a T-Loc. question for the same image may ask "What is rising from the water?" The latter question asks about a scene detail that is independent of the requested edit to the landmark's identity. Such questions often ask about image-specific visual content, which can change when backgrounds are modified or additional subjects are introduced. Therefore, we prompt Gemini to regenerate the reference answer for each T-Loc.

question using the corresponding image in each visual scenario. Consequently, the same locality question may have different reference answers under different visual scenarios.

A.1 METRICS CALCULATIONS

Here, we provide more details on the exact metric calculation. As a reminder, the VLM is denoted as f_θ, parametrized by θ. Given an image v and a textual query x, the VLM generates an output $y = f_\theta(v, x)$. The goal of knowledge editing is to update the model parameters to θ' using an editing dataset $\mathcal{D}_{edit}$, such that the edited model $f_{\theta'}$ outputs the target knowledge. For a specific editing instance in $\mathcal{D}_{edit}$, let (v_e, x_e, y_e^*) denote the editing triplet. All of our evaluation metrics are calculated based on the token-level accuracy of the generated sequence compared to the target sequence. To formalize this, we define an accuracy operator $\text{Acc}(\hat{y}, y)$. Let $y^{(k)}$ and $\hat{y}^{(k)}$ denote the k-th token of the target sequence y and the predicted sequence $\hat{y}$, respectively, and let $|y|$ be the total number of tokens in y. The accuracy operator is defined as:

$$\text{Acc}(\hat{y}, y) = \frac{1}{|y|} \sum_{k=1}^{|y|} \mathbb{I}[\hat{y}^{(k)} = y^{(k)}]$$

where $\mathbb{I}[\cdot]$ is the standard indicator function that equals 1 if the predicted token matches the target token, and 0 otherwise. Based on this operator, we formally define our metrics as follows:

Reliability (Rel.) measures whether the edited model $f_{\theta'}$ successfully outputs the target knowledge y_e^* when provided with the exact editing image and query:

$$M_{\text{Rel}} = \mathbb{E}_{(v_e, x_e, y_e^*) \sim \mathcal{D}_{edit}} [\text{Acc}(f_{\theta'}(v_e, x_e), y_e^*)]$$

Text Generality (T-Gen.) assesses the robustness of the edit against textual variations. Let $\mathcal{N}(x_e)$ represent a set of rephrased textual queries that share the same semantic meaning as x_e. The metric is defined as:

$$M_{\text{T-Gen.}} = \mathbb{E}_{(v_e, x_e, y_e^*) \sim \mathcal{D}_{edit}} \left[\mathbb{E}_{x \sim \mathcal{N}(x_e)} [\text{Acc}(f_{\theta'}(v_e, x), y_e^*)] \right]$$

Image Generality (I-Gen.) evaluates whether the injected knowledge successfully generalizes to unseen images containing the same subject. Let v_{iso}^{test} be the held-out clean image of the target subject. The metric is defined as:

$$M_{\text{I-Gen.}} = \mathbb{E}_{(v_e, x_e, y_e^*) \sim \mathcal{D}_{edit}} [\text{Acc}(f_{\theta'}(v_{iso}^{test}, x_e), y_e^*)]$$

Text Locality (T-Loc.) ensures that the editing process does not distort the model's original knowledge about the editing image when asked unrelated questions. Let x_{unrel} denote an unrelated textual query generated for the image v_e. The output of the edited model should maintain the accuracy relative to the original model's output:

$$M_{\text{T-Loc.}} = \mathbb{E}_{(v_e, x_e, y_e^*) \sim \mathcal{D}_{edit}} [\text{Acc}(f_{\theta'}(v_e, x_{unrel}), f_\theta(v_e, x_{unrel}))]$$

Image Locality (I-Loc.) ensures that the original knowledge remains intact when the edited model is presented with different images. Let $\mathcal{V}_{cat}$ represent a set of images belonging to the same category as v_e. We randomly sample a distracting image v_{other} from this set, strictly excluding the editing image itself (i.e., $v_{other} \in \mathcal{V}_{cat} \setminus \{v_e\}$), while keeping the textual query x_e identical:

$$M_{\text{I-Loc.}} = \mathbb{E}_{(v_e, x_e, y_e^*) \sim \mathcal{D}_{edit}} \left[\mathbb{E}_{v_{other} \sim \mathcal{V}_{cat} \setminus \{v_e\}} [\text{Acc}(f_{\theta'}(v_{other}, x_e), f_\theta(v_{other}, x_e))] \right]$$

A.2 DISTRACTEDIT CURATION

We show the Gemini prompt for curating DISTRACTEDIT.

Gemini prompt for curating DISTRACTEDIT

[SYSTEM MESSAGE]

You are a vision-language assistant that writes explicit edit instructions for a stronger image-editing model (Gemini).

Gemini prompt for curating DISTRACTEDIT (continued)

You receive: (1) the original image, (2) a subject name, (3) a category, and (4) a level description that defines what kind of change this edit should make.

Your job is to write SHORT EDIT INSTRUCTIONS, not a caption of the final image.

Every time you answer, you must:

- Look at the original image and consider the composition (for example: how large the subject appears in the frame, where the subject is located, how much empty space or background area is visible).
- Explicitly state what should stay the same in the original image (for example: 'Keep the original subject...', 'Do not change the subject's pose...').
- Explicitly state what should be changed, replaced, or added (for example: 'Change the background to...', 'Replace the pose with...', 'Add effects that...').
- Follow the level description as a hard constraint for what is allowed to change.
- Be reasonably concise (1–3 sentences) and focused on instructions.
- Do NOT write a pure caption of the final image.
- Do NOT mention 'level', 'category', or any meta-instructions.

In addition, you may optionally add ONE short composition note for the editing model:

- If helpful, append one extra sentence starting with 'Note for Gemini:' that gives a high-level hint based on the original composition (for example: the subject fills most of the frame so the background should stay simple, or the subject is small so a more detailed background is acceptable).

Bad example (caption style):

"A person wearing a red jacket stands in front of a night-time city skyline with bright neon lights."

Good example (edit-instruction style):

"Keep the original person exactly as they are, including the red jacket and pose. Change only the background so they stand in front of a night-time city skyline with bright neon lights. Note for Gemini: the person fills most of the frame, so keep most background detail toward the edges behind them."

Always answer in the good style above: explicit keep/change/add instructions, not captions. Return only the edit instructions (and optionally one 'Note for Gemini' sentence).

[EDIT INSTRUCTION TEMPLATE]

We are editing an existing image of a known subject.

Subject name: {`subject`}
Category: {`category`}
Edit level description for this category:
{`level_template`}

Task:

- Look at the current (original) image of the subject, including how large the subject appears, where they are positioned in the frame, and how much background is visible.
- Use the edit level description as a strict rule for what may change and what must stay similar to the original image.
- Write 1–3 sentences of explicit edit instructions for another image-editing model.
- Your instructions MUST clearly mention:
  - what to KEEP or NOT CHANGE in the original image, and
  - what to CHANGE, REPLACE, or ADD (such as background, pose, viewpoint, occlusion, or effects), following the description.
- Use verbs like 'Keep...', 'Do not change...', 'Change...', 'Replace...', 'Add...', rather than just describing the final image.
- Optionally, if helpful, append ONE extra sentence starting with 'Note for Gemini:' that gives a brief hint about composition or proportions (for example: the subject is very large in the frame so the new background should not be too detailed behind them).

Gemini prompt for curating DISTRACTEDIT (continued)

- Do NOT write a pure caption of the final image.
- Do NOT mention 'level', 'category', 'template', or these instructions.

Return only the edit instructions (and optionally one 'Note for Gemini' sentence):

The level templates are the scripts that we include to provide general scene guidance for each category.

A.3 IMPLEMENTATION DETAILS OF EDITING METHODS

We use AdamW for all KE methods and two NVIDIA A6000 GPUs. MEND and VisEdit require an additional training set, where we use the 800 samples from Editing VQA Cheng et al. (2023) to train the model. The reason that VisEdit is only evaluated under Blip2-OPT 2.7B is that the current implementation of VisEdit only supports a fixed number of visual tokens. However, the number of visual tokens is dynamically allocated based on the size of the input images under InternVL-3.5 8B.

FT-L updates model parameters by performing gradient descent on the chosen model parameters, where "L" stands for Language Model, meaning the parameters are located in a specific layer in the Language Model of the corresponding VLM.

LoRA updates the model by freezing the pre-trained weights and injecting trainable low-rank decomposition matrices into the chosen parameters.

IKE retrieves similar demonstrations from an additional dataset to inject new knowledge via prompting. Similarly, we employ Editing VQA. This procedure is uniformly applied across all evaluated models.

MEND trains lightweight auxiliary networks, which execute precise, localized modifications using just a single input-output pair.

GRACE leaves the pre-trained model weights intact. Instead, it implements targeted spot-fixes by writing new mappings into a discrete, local codebook within the model's latent space.

BalancEdit dynamically generates positive and negative samples to accurately determine the specific scope of an edit.

VisEdit dynamically corrects erroneous or outdated knowledge by editing these intermediate visual representations in prompt-critical regions.

Table 5: Comparison of Edit Layers and Hyperparameters across Methods and Models

Method	Model	(Max) Steps	Edit Layer	LR
FT	BLIP2 2.7B	10	29th language model layer	8e-5
FT	InternVL 3.5 8B	85	25th language model layer	1e-4
LoRA	BLIP2 2.7B	15	29th language model layer	3e-3
LoRA	InternVL 3.5 8B	85	30th language model layer	3e-3
MEND	BLIP2 2.7B	–	29th–31st language model layers	1e-4
MEND	InternVL 3.5 8B	–	26th–28th language model layers	1e-4
WISE	BLIP2 2.7B	10	23rd language model layer	1.0
WISE	InternVL 3.5 8B	5	30th language model layer	1e-3
GRACE	BLIP2 2.7B	20	31st language model layer	1.0
GRACE	InternVL 3.5 8B	20	15th language model layer	1.0
BalancEdit	BLIP2 2.7B	8	29th language model layer	0.01
BalancEdit	InternVL 3.5 8B	50	30th language model layer	0.01
VisEdit	BLIP2 2.7B	8	19th language model layer	1e-4
VisEdit	InternVL 3.5 8B	-	-	-

A.4 THE CRITERIA FOR HUMAN ANNOTATIONS

The three annotators independently evaluated each generated candidate image using a five-criterion checklist:

1. **Scenario compliance:** The image follows the intended visual-distraction scenario, i.e., *Complex Scene* or *Multi-Subj*.
2. **Target entity preservation:** The original target entity remains clearly identifiable, and the image does not introduce ambiguity about which entity is the editing target.
3. **Distraction validity:** The added background or co-occurring entities are plausible and relevant to the intended distraction type. For example, in the *Multi-Subj* scenario, the additional entities should be visually similar or semantically related to the target entity.
4. **Photorealism and physical plausibility:** The image looks like a plausible real-world photo, with consistent lighting, shadows, perspective, scale, and geometry.
5. **Benchmark diversity:** Compared with other accepted images, the candidate should not be overly similar in visual elements such as style, composition, viewpoint, or distraction patterns.

Each annotator assigns a score from 0 to 5 and gives a final accept/reject decision. For each knowledge seed and scenario, we generate multiple candidate images and select the valid candidate with the highest average score. Candidates are discarded if their average score is below 4 or if the annotators' scores differ by more than two points on any criterion. Approximately 25% of generated candidates pass this filtering stage and are included in DISTRACTEDIT.

A.5 CONSISTENCY STUDY

To examine whether the findings on DISTRACTEDIT extend to real-world images, we conduct a small-scale consistency study using 20 knowledge seeds. For each seed, we collect candidate images through web crawling and filter them according to our visual scenario definitions. Specifically, *Real Complex Scene* contains the target subject in a cluttered background, while *Real Multi-Subj* additionally includes visually similar or semantically related subjects. Three human annotators filter and validate the collected images following the applicable criteria in App. A.4.

We evaluate the editing methods and VLM backbones considered in Sec. 3.2, using the metrics defined in App. A.1. All three conditions use the same 20 knowledge seeds: *Original* uses their isolated-entity images, while the two real-world conditions use the collected images for editing.

Consistent degradation in image generality. Tab. 6 shows that the main trend observed on DISTRACTEDIT also appears on real-world images. **I-Gen.** declines for most methods under both distraction scenarios. For example, WISE on InternVL drops from 0.91 under *Original* to 0.69 and 0.67 under *Real Complex Scene* and *Real Multi-Subj*, respectively. These declines can occur even when editing reliability remains high. BalancEdit on InternVL maintains a **Rel.** of 1.00 across all three conditions, while its **I-Gen.** drops from 0.85 to 0.56 and 0.60.

Mechanism- and backbone-specific effects. The degradation patterns also vary across editing methods and backbones, consistent with Sec. 3.2. IKE, which updates knowledge through in-context learning, preserves **I-Gen.** on InternVL at 0.58, while its **T-Loc.** decreases from 0.88 to 0.85 and 0.82. On Blip2, its **I-Gen.** even improves under *Real Multi-Subj*, but **I-Loc.** drops from 0.35 to 0.29. Thus, its degradation is primarily reflected in locality. Backbone differences are also evident: under *Real Complex Scene*, MEND's **I-Gen.** decreases from 1.00 to 0.87 on InternVL but increases from 0.87 to 0.92 on Blip2.

Variation across distraction types. *Real Multi-Subj* often exhibits larger **I-Gen.** declines than *Real Complex Scene*, as illustrated by GRACE on Blip2 (0.86 to 0.81 and 0.63). However, this ordering is not universal: BalancEdit on InternVL performs slightly better under *Real Multi-Subj* than under *Real Complex Scene*, although both remain below *Original*.

These results provide supporting evidence that the central findings of DISTRACTEDIT also hold on real-world images. Since this study covers only 20 knowledge seeds and naturally occurring images vary in appearance, viewpoint, and image quality, it complements the controlled comparisons in DISTRACTEDIT.

Method	Metric	InternVL-3.5-8B			Blip2-OPT 2.7B		
		Original	Real Complex Scene	Real Multi-Subj	Original	Real Complex Scene	Real Multi-Subj
FT-L	Rel.	1.00	$1.00_{+0.0\%}$	$1.00_{+0.0\%}$	0.87	$0.91_{+4.6\%}$	$0.93_{+6.9\%}$
	T-Gen.	0.80	$0.92_{+15.0\%}$	$0.86_{+7.5\%}$	0.64	$0.63_{-1.6\%}$	$0.58_{-9.4\%}$
	I-Gen.	0.91	$0.75_{-17.6\%}$	$0.66_{-27.5\%}$	0.73	$0.58_{-20.5\%}$	$0.58_{-20.5\%}$
	T-Loc.	0.97	$1.00_{+3.1\%}$	$0.95_{-2.1\%}$	0.85	$0.93_{+9.4\%}$	$0.88_{+3.5\%}$
	I-Loc.	0.86	$0.90_{+4.7\%}$	$0.95_{+10.5\%}$	0.83	$0.86_{+3.6\%}$	$0.85_{+2.4\%}$
	Avg.	0.91	$0.91_{+0.7\%}$	$0.88_{-2.6\%}$	0.78	$0.78_{-0.3\%}$	$0.76_{-2.6\%}$
IKE	Rel.	0.58	$0.58_{+0.0\%}$	$0.58_{+0.0\%}$	0.93	$0.93_{+0.0\%}$	$0.98_{+5.4\%}$
	T-Gen.	0.59	$0.58_{-1.7\%}$	$0.58_{-1.7\%}$	0.48	$0.48_{+0.0\%}$	$0.48_{+0.0\%}$
	I-Gen.	0.58	$0.58_{+0.0\%}$	$0.58_{+0.0\%}$	0.93	$0.93_{+0.0\%}$	$0.98_{+5.4\%}$
	T-Loc.	0.88	$0.85_{-3.4\%}$	$0.82_{-6.8\%}$	1.00	$0.93_{-7.0\%}$	$0.90_{-10.0\%}$
	I-Loc.	0.75	$0.75_{+0.0\%}$	$0.75_{+0.0\%}$	0.35	$0.35_{+0.0\%}$	$0.29_{-17.1\%}$
	Avg.	0.68	$0.67_{-1.2\%}$	$0.66_{-2.1\%}$	0.74	$0.72_{-1.9\%}$	$0.73_{-1.6\%}$
MEND	Rel.	0.98	$0.93_{-5.1\%}$	$0.90_{-8.2\%}$	0.94	$0.95_{+1.1\%}$	$0.95_{+1.1\%}$
	T-Gen.	0.95	$0.88_{-7.4\%}$	$0.90_{-5.3\%}$	0.80	$0.78_{-2.5\%}$	$0.76_{-5.0\%}$
	I-Gen.	1.00	$0.87_{-13.0\%}$	$0.86_{-14.0\%}$	0.87	$0.92_{+5.7\%}$	$0.85_{-2.3\%}$
	T-Loc.	0.76	$0.75_{-1.3\%}$	$0.90_{+18.4\%}$	0.50	$0.55_{+10.0\%}$	$0.50_{+0.0\%}$
	I-Loc.	0.75	$0.81_{+8.0\%}$	$0.78_{+4.0\%}$	0.69	$0.69_{+0.0\%}$	$0.70_{+1.4\%}$
	Avg.	0.89	$0.85_{-4.5\%}$	$0.87_{-2.3\%}$	0.76	$0.78_{+2.4\%}$	$0.75_{-1.1\%}$
LoRA	Rel.	1.00	$1.00_{+0.0\%}$	$1.00_{+0.0\%}$	0.87	$0.90_{+3.4\%}$	$0.92_{+5.7\%}$
	T-Gen.	0.81	$0.87_{+7.4\%}$	$0.80_{-1.2\%}$	0.72	$0.80_{+11.1\%}$	$0.68_{-5.6\%}$
	I-Gen.	0.85	$0.70_{-17.6\%}$	$0.70_{-17.6\%}$	0.86	$0.82_{-4.7\%}$	$0.77_{-10.5\%}$
	T-Loc.	0.97	$1.00_{+3.1\%}$	$1.00_{+3.1\%}$	0.68	$0.72_{+5.9\%}$	$0.68_{+0.0\%}$
	I-Loc.	0.92	$0.93_{+1.1\%}$	$0.97_{+5.4\%}$	0.78	$0.80_{+2.6\%}$	$0.80_{+2.6\%}$
	Avg.	0.91	$0.90_{-1.1\%}$	$0.89_{-1.8\%}$	0.78	$0.81_{+3.3\%}$	$0.77_{-1.5\%}$
WISE	Rel.	1.00	$0.94_{-6.0\%}$	$0.92_{-8.0\%}$	0.82	$0.83_{+1.2\%}$	$0.87_{+6.1\%}$
	T-Gen.	0.94	$0.87_{-7.4\%}$	$0.81_{-13.8\%}$	0.60	$0.54_{-10.0\%}$	$0.55_{-8.3\%}$
	I-Gen.	0.91	$0.69_{-24.2\%}$	$0.67_{-26.4\%}$	0.81	$0.66_{-18.5\%}$	$0.67_{-17.3\%}$
	T-Loc.	0.95	$1.00_{+5.3\%}$	$0.90_{-5.3\%}$	0.68	$0.68_{+0.0\%}$	$0.57_{-16.2\%}$
	I-Loc.	0.81	$0.98_{+21.0\%}$	$0.93_{+14.8\%}$	0.56	$0.56_{+0.0\%}$	$0.64_{+14.3\%}$
	Avg.	0.92	$0.90_{-2.8\%}$	$0.85_{-8.2\%}$	0.69	$0.65_{-5.8\%}$	$0.66_{-4.9\%}$
GRACE	Rel.	1.00	$0.97_{-3.0\%}$	$1.00_{+0.0\%}$	0.92	$0.90_{-2.2\%}$	$0.83_{-9.8\%}$
	T-Gen.	0.88	$0.86_{-2.3\%}$	$0.88_{+0.0\%}$	0.78	$0.76_{-2.6\%}$	$0.65_{-16.7\%}$
	I-Gen.	0.64	$0.58_{-9.4\%}$	$0.51_{-20.3\%}$	0.86	$0.81_{-5.8\%}$	$0.63_{-26.7\%}$
	T-Loc.	0.95	$0.93_{-2.1\%}$	$0.95_{+0.0\%}$	0.95	$0.95_{+0.0\%}$	$0.95_{+0.0\%}$
	I-Loc.	0.88	$0.90_{+2.3\%}$	$0.97_{+10.2\%}$	0.75	$0.95_{+26.7\%}$	$0.93_{+24.0\%}$
	Avg.	0.87	$0.85_{-2.5\%}$	$0.86_{-0.9\%}$	0.85	$0.87_{+2.6\%}$	$0.80_{-6.3\%}$
BalanceEdit	Rel.	1.00	$1.00_{+0.0\%}$	$1.00_{+0.0\%}$	0.87	$0.91_{+4.6\%}$	$0.93_{+6.9\%}$
	T-Gen.	0.93	$0.97_{+4.3\%}$	$1.00_{+7.5\%}$	0.65	$0.63_{-3.1\%}$	$0.58_{-10.8\%}$
	I-Gen.	0.85	$0.56_{-34.1\%}$	$0.60_{-29.4\%}$	0.73	$0.60_{-17.8\%}$	$0.58_{-20.5\%}$
	T-Loc.	0.95	$0.93_{-2.1\%}$	$0.90_{-5.3\%}$	0.85	$0.93_{+9.4\%}$	$0.88_{+3.5\%}$
	I-Loc.	0.95	$1.00_{+5.3\%}$	$0.95_{+0.0\%}$	0.93	$0.93_{+0.0\%}$	$0.95_{+2.2\%}$
	Avg.	0.94	$0.89_{-4.7\%}$	$0.89_{-4.9\%}$	0.81	$0.80_{-0.7\%}$	$0.78_{-2.7\%}$
VisEdit	Rel.	–	–	–	0.93	$0.90_{-3.2\%}$	$0.90_{-3.2\%}$
	T-Gen.	–	–	–	0.93	$0.91_{-2.2\%}$	$0.94_{+1.1\%}$
	I-Gen.	–	–	–	0.94	$0.91_{-3.2\%}$	$0.90_{-4.2\%}$
	T-Loc.	–	–	–	0.95	$0.95_{+0.0\%}$	$0.95_{+0.0\%}$
	I-Loc.	–	–	–	0.28	$0.28_{+0.0\%}$	$0.28_{+0.0\%}$
	Avg.	–	–	–	0.81	$0.79_{-2.0\%}$	$0.79_{-1.6\%}$

Table 6: Editing performance on real-world images. For 20 knowledge seeds, the editing image is replaced by a real-world photo with a complex scene (Real Complex Scene) or multiple subjects (Real Multi-Subj); Original uses the isolated-entity images of the same seeds. Degraded scores are highlighted in red.

B More details on SACO

B.1 More details on quantitative results of SACO

Table 7 reports the complete results of SACO on InternVL-3.5-8B across different scenarios. The results confirm that the findings discussed in Sec. 5.1 hold across both the Complex Scene and Multi-Subj settings. Across the evaluated editing methods, SACO consistently improves image generality. The effects on other metrics remain method- and scenario-dependent. Improvements in image generality can have tradeoffs in locality or text generality. These observations reinforce our discussion in Sec. 5.1 that visual refinement improves the generalization of edited knowledge to unseen images, while precise control over the scope of knowledge editing remains a challenge.

B.2 More details on ablation study of SACO

We estimate the cost per query in US dollars by applying the official Gemini 3 API pricing based on the returned response. We measure latency using a wall-clock timer that records the time from the start of processing each query to its completion, and report it in seconds.

Method	Metric	*BoA*	Original		Complex Scene		Multi-Subj	
			Raw Visual	Refined Visual	Raw Visual	Refined Visual	Raw Visual	Refined Visual
MEND	**Rel.**	*0.91*	0.91	0.87	0.83	0.87	0.81	0.87
	T-Gen.	*0.87*	0.87	0.87	0.83	0.87	0.84	0.87
	I-Gen.	*0.89*	0.89	0.84	0.79	0.85	0.72	0.81
	T-Loc.	*0.90*	0.90	0.87	0.88	0.85	0.87	0.87
	I-Loc.	*0.77*	0.74	0.78	0.74	0.74	0.77	0.77
	Avg.	*0.87*	0.86	0.85	0.81	0.84	0.80	0.84
WISE	**Rel.**	*1.00*	1.00	1.00	0.97	1.00	0.99	1.00
	T-Gen.	*0.84*	0.84	0.84	0.84	0.85	0.83	0.86
	I-Gen.	*0.97*	0.97	0.97	0.71	0.97	0.47	0.90
	T-Loc.	*0.98*	0.98	0.98	0.98	0.98	0.98	0.95
	I-Loc.	*0.96*	0.71	0.68	0.93	0.71	0.96	0.71
	Avg.	*0.95*	0.90	0.90	0.89	0.90	0.85	0.88
GRACE	**Rel.**	*1.00*	1.00	1.00	1.00	1.00	0.99	1.00
	T-Gen.	*0.78*	0.77	0.79	0.77	0.78	0.78	0.78
	I-Gen.	*0.65*	0.65	0.66	0.53	0.67	0.46	0.60
	T-Loc.	*0.93*	0.93	0.95	0.93	0.95	0.91	0.92
	I-Loc.	*0.89*	0.78	0.75	0.83	0.73	0.89	0.89
	Avg.	*0.85*	0.83	0.83	0.81	0.83	0.81	0.84
BalancEdit	**Rel.**	*1.00*	1.00	1.00	1.00	1.00	1.00	1.00
	T-Gen.	*0.98*	0.90	0.90	0.98	0.91	0.97	0.96
	I-Gen.	*0.97*	0.97	0.97	0.82	0.97	0.63	0.93
	T-Loc.	*0.98*	0.98	0.98	0.97	0.98	0.94	0.95
	I-Loc.	*0.91*	0.65	0.64	0.84	0.65	0.91	0.65
	Avg.	*0.97*	0.90	0.90	0.92	0.90	0.89	0.90

Table 7: Detailed performance of SACO visual refinement on InternVL-3.5-8B across the three visual scenarios. ***BoA*** reports the *best-of-all* performance across visual scenarios before refinement, following Table 2. Raw and Refined Visual denote the visual inputs before and after applying SACO, respectively.

Table 8 shows that using only half of the analysis, through either SACO-FE or SACO-DR, reduces the latency but only provides modest cost savings compared to full SACO. Replacing generative image editing with cropping (SACO-IC) substantially reduces cost and also lowers latency. Combined with the performance results in Tab. 3, cropping is competitive for some editing methods. These observations motivate a future direction of SACO with multiple image-processing tools, which allows the agent to select or combine different tools to suit the visual input and editing request. Such adaptive tool use could offer a better balance between editing performance, cost, and latency.

Variant	Cost per query ($)	Latency per query (s)
SACO-FE	0.069	55.38
SACO-DR	0.071	59.04
SACO-IC	0.004	57.74
SACO	0.072	80.29

Table 8: Per-query cost and latency of SACO and its ablated variants.

Fig. 7 compares the raw visuals with the images produced by SACO-FE, SACO-DR, SACO-IC, and full SACO. These examples illustrate the different effects of the ablated variants on the target subject and its surrounding context. SACO-FE refines the presentation of the target, for example by making its face more directly visible, while much of the original scene and the competing subjects remain. SACO-DR removes distracting subjects or backgrounds, but does not consistently increase the target's prominence. In the hare example, the other animals are removed, while the target still occupies a small region within the grassy scene. SACO-IC produces a closer view of the target and retains its original visual details. These visual patterns illustrate the complementary roles of focus enhancement and distraction removal, consistent with the stronger overall performance of full SACO.

B.3 PROMPTS FOR SACO

We include prompts for SACO:



Figure 7: Visual demonstration of SACO and its ablated variants. From left to right: raw visual input, focus enhancement only (SACO-FE), distraction removal only (SACO-DR), image cropping (SACO-IC), and full SACO.

Knowledge Perception Agent Prompt

[SYSTEM PROMPT]

You are a **Strategic Visual Knowledge Agent** serving as the analytical core of a Vision-Language Model (VLM) Knowledge Editing pipeline.

The Generalization Problem

Your objective is to prevent VLM overfitting during knowledge editing. When a VLM is updated using a specific image, it inherently suffers from "lazy learning." It will memorize the most prominent visual tokens—the shape of the host object, the background, the lighting, or the surrounding context—rather than the actual knowledge concept.

If the true knowledge is a brand logo, but you feed the VLM a photo of a plastic device displaying that logo, the VLM will associate the knowledge with the plastic device. Generalization fails.

Your Mandate

You must diagnose the provided image and `source_knowledge` to identify the **Minimal Visual Carrier**: the absolute smallest, purest arrangement of visual features that unambiguously identifies the concept. You must then devise a strategy to radically isolate this carrier, stripping away all host objects, environments, and stylistic noise that pose an overfitting risk.

Output Structure

Use the `<reasoning>` block as your unstructured analytical scratchpad to diagnose the image and solve the Generalization Problem. Following your reasoning, output a strict JSON object that captures your final strategic blueprint for the downstream generator.

```
<reasoning>
[Analyze the image and the source knowledge. Diagnose the overfitting
     risks. Determine the minimal visual carrier and formulate your
    isolation strategy.]
</reasoning>
FINAL_JSON_OUTPUT:
{
  "minimal_visual_carrier": "Precise description of the exact visual
    features that hold the true identity.",
  "isolation_strategy_name": "A descriptive name for your custom
    extraction approach (e.g., 'Flat Logo Extraction', 'Facial
    Geometry Isolation').",
```

Knowledge Perception Agent Prompt (continued)

```
  "overfitting_traps_avoided": [
    {
      "rejected_visual_element": "A specific element from the input
    image you chose to discard.",
      "overfitting_risk": "Why including this element would cause the
     VLM to learn a spurious correlation."
    }
  ],
  "reconstruction_blueprint": {
    "core_subject": "Exactly what the downstream generator should
    render.",
    "presentation_format": "The required visual format (e.g., 'Flat 2
    D vector graphic', 'Neutral photographic portrait', 'Isolated 3D
    geometry'). Do not assume 3D/lighting/framing unless the concept
    strictly requires it.",
    "explicit_exclusions": [
      "List of specific things (contexts, host objects, backgrounds)
    that the generator MUST NOT include."
    ]
  },
  "residual_risks": "Any remaining ambiguities the downstream planner
     should be aware of.",
  "reasoning_summary": "A 3-5 sentence plain-language summary of your
     core reasoning: what the minimal visual carrier is, why the main
    overfitting traps were rejected, and what the downstream generator
     must do."
}
```

[USER MESSAGE FORMAT]

Knowledge Editing Context

The ultimate goal of this pipeline is to update the VLM's text output based on the provided image. However, your specific job right now is to focus EXCLUSIVELY on isolating the visual carrier of the current concept. Do not worry about what the knowledge will be changed into.

- **Source Knowledge (The Concept to Isolate):** {`source_knowledge`}

Instructions

Your analysis determines whether the downstream edit will successfully generalize or catastrophically overfit.

1. Diagnose the image and formulate your strategy within the `<reasoning>` block.
2. Output the phrase `FINAL_JSON_OUTPUT:` on a new line.
3. Output the raw JSON object detailing your reconstruction blueprint.

Planner Agent Prompt

[SYSTEM PROMPT]

You are a **Canonical Image Editing Planner Agent** in an iterative Vision-Language Model (VLM) knowledge editing pipeline.

Your responsibility is to synthesize the "Canonicalization Analysis" from the Perception Agent (which identifies the core conceptual anchor and the instance biases to strip) WITH your own direct visual observation of the input image. You will translate this multimodal understanding into a vivid, highly detailed text instruction for a downstream Image Editing/Generation model.

Core Directives

1. **Multimodal Grounding:** Do not rely solely on the text JSON. You MUST look at the provided input image to see the current state of the `minimal_visual_carrier` and visually locate

Planner Agent Prompt (continued)

the `overfitting_traps_avoided`. Use the image to add concrete, vivid details to your reconstruction plan.

2. **Execute Reconstruction Blueprint:** Your primary goal is to break the specificity of the input image to maximize VLM generalization. You must generate a prompt that explicitly commands the editing model to follow the `reconstruction_blueprint` — stripping all rejected visual elements and rendering only the `minimal_visual_carrier` in the specified `presentation_format`.
3. **Action-Oriented & Descriptive Prompting:** Your final prompt must be an explicit, vivid instruction. Tell the model exactly how to pose, light, and frame the core concept. For example, instead of just "keep the logo", write "Transform the device to a direct, front-facing angle with flat studio lighting, ensuring the central logo is sharply in focus and highly contrasted against a pure white background."
4. **Focus on the Anchor:** Ensure the defining features of the anchor are front and center, clearly resolved, and completely free of any occlusions or confusing context.

Output Format Strict Requirements

You must structure your response with a detailed reasoning block followed by a strictly formatted JSON object.

```
<reasoning>
[Step 1: Visual Grounding - Locate the `minimal_visual_carrier` in
    the current image and visually identify the elements listed in `
    overfitting_traps_avoided`.
Step 2: Isolation Strategy - Formulate how to instruct the editing
    model to execute the `reconstruction_blueprint`, removing all
    rejected elements and rendering only the carrier in the specified
    `presentation_format`.
Step 3: Prompt Synthesis - Draft the exact wording to ensure the
    final image is free of overfitting traps and represents the
    generalized, "Platonic ideal" of the concept.
Be as concise as possible; only elaborate when necessary.]
</reasoning>
FINAL_JSON_OUTPUT:
{
  "edit_instruction": "The exact, vivid, and highly detailed
    instruction string that will be fed into the Image Editing model
    to produce the canonical representation.",
  "reasoning_summary": "A 3-5 sentence plain-language summary of your
     core reasoning: what you observed in the image, how it maps to
    the reconstruction blueprint, and why your edit instruction
    achieves the canonical representation."
}
```

[USER MESSAGE FORMAT]

Perception Analysis Context (Canonicalization Plan)

```
```json
{perception_json}
```
```

Instructions

1. Carefully review the Perception Analysis JSON AND visually inspect the accompanying input image.
2. Think step-by-step inside the `<reasoning>` tags. Ground the conceptual anchor and biases in the actual visual details of the image.
3. Plan a precise editing instruction that strips away the instance biases and executes the canonical transformation, resulting in a generalized, highly recognizable representation.
4. Immediately after closing the reasoning block, output the exact phrase `FINAL_JSON_OUTPUT:` on a new line.

Planner Agent Prompt (continued)

5. Directly below that phrase, output the raw JSON object containing only your highly descriptive `edit_instruction`.