LOCATING AND EDITING KNOWLEDGE-CONCERNED NEURONS IN VISION LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge editing (KE) enables large language models (LLMs) to efficiently correct outdated or erroneous information without costly retraining. Recently, locate-then-edit has emerged as an effective paradigm due to its efficiency and interpretability, offering an alternative to memory-based approaches and methods that require training auxiliary editors. However, in vision large language models (VLLMs), factual knowledge is encoded across both visual and textual representations, making knowledge localization fundamentally different from that in text-only LLMs. Existing multimodal knowledge editing methods, such as MKE, typically rely on additional training, which may introduce out-of-distribution challenges. To address this gap, we propose DELTA (Δ), to the best of our knowledge, the first fine-grained neuron-level locate-then-edit framework that specially designed for multimodal knowledge editing that requires no auxiliary editor optimization. DELTA first identifies candidate layers by jointly modeling cross-modal fusion, feed-forward network engagement, and activation traceability. It then localizes knowledge-relevant neurons through attribution scores and visual filtering. For knowledge updating, DELTA constrains parameter changes toward an approximate null space to preserve unrelated knowledge, while perturbing visual embeddings to account for diverse visual evidence. Extensive experiments demonstrate that DELTA consistently outperforms existing knowledge editing approaches, highlighting the effectiveness of locate-then-edit for VLLMs and providing a step toward general-purpose multimodal knowledge editing.

1 INTRODUCTION

Large Language Models (LLMs) and Vision Large Language Models (VLLMs) (Li et al., 2023; Liu et al., 2023; Zhu et al., 2024; Hong et al., 2026; Bai et al., 2025; Lu et al., 2024) have emerged as the dominant paradigm for artificial intelligence, substantially extending unified perception, reasoning, and instruction-following capabilities across multimodal inputs. Their remarkable performance has driven significant advances in both natural language processing and computer vision, enabling a wide range of real-world applications. Despite this impressive progress, the knowledge encoded in these models inevitably becomes outdated over time and may also contain inaccuracies, biases, and toxic content inherited from large-scale pretraining corpora. Since full retraining of LLMs is computationally prohibitive, knowledge editing (KE) has emerged as a lightweight alternative that updates specific knowledge while preserving overall capabilities.

Recent research on knowledge editing mainly focuses on LLMs with memory-based (Mitchell et al., 2022b; Kassner et al., 2021; Huang et al., 2023; Dong et al., 2022; De Cao et al., 2021; Mitchell et al., 2022a; Ni et al., 2024; Li et al., 2025) or training-based editors (De Cao et al., 2021; Mitchell et al., 2022a; Ni et al., 2024; Li et al., 2025) while struggling with multimodal settings. Among LLM editing approaches, the locate-then-edit (Meng et al., 2022; 2023; Li et al., 2024; Pan et al., 2025; Jiang et al., 2025; Fang et al., 2025) paradigm is particularly attractive: it first identifies factual knowledge stored in specific neurons or layers, typically via causal tracing (Meng et al., 2022), and then modifies them directly without any additional editor training. This yields a general-purpose, interpretable editor. In essence, factual associations within LLMs are predominantly triggered by a small set of language subject tokens, making causal tracing effective. However, extending these methods to VLLMs confronts challenges, where subject information is jointly expressed through text tokens and a large grid of visual tokens. The sheer number of visual tokens (often hundreds)

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.

washes out the fine-grained variability required for reliable causal attribution. Furthermore, since cross-modal attention continually fuses encoded visual evidence into the hidden states of text tokens, the factual signal is no longer confined to any single token but distributed across multimodal representations, which makes localization more challenging on VLLMs than on LLMs.

Current multimodal knowledge editing (MKE) (Wang et al., 2024; Chen et al., 2025b;a; Pan et al., 2024b; Su et al., 2026) is still in its infancy and primarily relies on training additional parameters or editors. Though these methods fit multimodal editing patterns to some extent, we observe a critical weakness: their editing capability degrades sharply when faced with out-of-distribution (OOD) examples (e.g., Tab. 1 showcases an accuracy drop over 20%). This suggests that current training-required editors are likely to learn how to edit on specific datasets only, rather than directly injecting target knowledge into VLLMs. All these lead to a fundamental gap: currently there is no *locate-then-edit* method that edits multimodal knowledge by interpreting internal signals within VLLMs.

Motivated by this, we propose **DELTA**(Δ), a **d**ecoupled vision-t**e**xt **l**ocalization and edi**t**ing framework via neuron **a**ttribution for knowledge editing in vision large language models. Fig. 1 displays the pipeline of our DELTA, which is the *first* locate-then-edit knowledge editing approach for VLLMs, to the best of our knowledge. First, we propose a layer-wise condition to pick candidate layers satisfying three criteria, where (C1)–*Fusion* captures the cross-modal fusion level within each layer, (C2)–*Engagement* measures the involvement of feed-forward network (FFN) layers, and (C3)–*Tracibility* quantifies the entropy of normalized neuronal activations to identify layers where responses are broadly distributed across units. Next, we localize knowledge-concerned neurons with the top-K attribution scores within candidate layers and leverage a visually dominant filtering to obtain robust multimodal editing targets, treating the remaining neurons as the final editing targets. To update multimodal knowledge, we additionally constrain the parameter updates to the approximate null-space by optimizing their projection onto the direct sum space. Furthermore, visual perturbations are applied to visual embeddings to adapt the diversity of visual evidence to the consistent multimodal information.

Extensive experiments on different VLLM backbones such as BLIP2-OPT-2.7B (Li et al., 2023), LLaVA-v1.5-7B (Liu et al., 2023), and MiniGPT4-Vicuna-7B (Zhu et al., 2024) demonstrate the effectiveness of **DELTA**, outperforming other training-required and locate-then-edit approaches with no dependence on specific data distributions or training requirements. Ablation studies exhibit the effectiveness of the core components, and show trade-offs among different metrics by varying parameters. We find that previous locate-then-edit approaches proposed for LLMs also benefit from the localization and editing strategies provided by our **DELTA**.

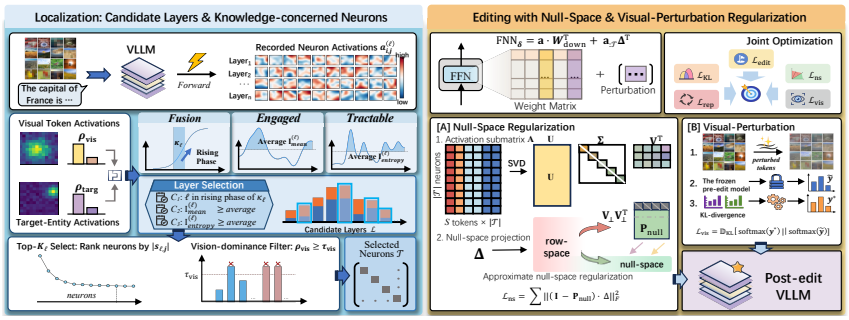

Figure 1: Overview of **DELTA** framework. We first denote activations in the forward process and then analyze the variation tendency via the decoupled vision–text activations to pick the candidate layers and target neurons. When updating knowledge, we introduce null-space regularized optimization and visual perturbation for modality locality preservation.

2 Preliminaries

2.1 Vision Large Language Models

Decoder-only (Vig & Belinkov, 2019; Vaswani et al., 2017) VLLMs typically consist of a vision transformer (ViT) (Dosovitskiy et al., 2021). Let $f_{\mathcal{W}}$ be a VLLM with L transformer layers, hidden dimension d_h, FFN intermediate dimension d_{ffn}. An image-text pair query $(\boldsymbol{I}, \boldsymbol{p})$ is embedded

by a vision encoder and mapped by a projector to N_v vision-token embeddings $\boldsymbol{z}_{\text{vis}} \in \mathbb{R}^{N_v \times d_h}$ concatenated with N_t text-token embeddings $\boldsymbol{z}_{\text{txt}} \in \mathbb{R}^{N_t \times d_h}$ to form the initial hidden state $\boldsymbol{h}^{(0)} = [\boldsymbol{z}_{\text{vis}}; \boldsymbol{z}_{\text{txt}}] \in \mathbb{R}^{S \times d_h}$, where $S = N_v + N_t$. Layer $\ell \in [1, L]$ updates the residual stream by $\boldsymbol{h}^{(\ell)} = \boldsymbol{h}^{(\ell-1)} + \text{Attn}^{(\ell)}(\boldsymbol{h}^{(\ell-1)}) + \text{FFN}^{(\ell)}(\tilde{\boldsymbol{h}}^{(\ell-1)})$, where $\tilde{\boldsymbol{h}}^{(\ell-1)} = \boldsymbol{h}^{(\ell-1)} + \text{Attn}^{(\ell)}(\boldsymbol{h}^{(\ell-1)})$. The FFN block at layer ℓ is an MLP with up and down projection weights $\mathbf{W}_{\text{up}}^{(\ell)} \in \mathbb{R}^{d_h \times d_{\text{ffn}}}$ and $\mathbf{W}_{\text{down}}^{(\ell)} \in \mathbb{R}^{d_h \times d_{\text{ffn}}}$, internal layer norm $\gamma(\cdot)$, and FFN non-linearity $\sigma_{\text{ffn}}(\cdot)$, formally:

$$\text{FFN}^{(\ell)}(\tilde{\boldsymbol{h}}^{(\ell-1)}) = \mathbf{a}^{(\ell)}\, \mathbf{W}_{\text{down}}^{(\ell)\ \top} \in \mathbb{R}^{S \times d_h}, \tag{1}$$

where $\mathbf{a}^{(\ell)} = \sigma_{\text{ffn}}\big(\gamma(\tilde{\boldsymbol{h}}^{(\ell-1)})\, \mathbf{W}_{\text{up}}^{(\ell)}\big) \in \mathbb{R}^{S \times d_{\text{ffn}}}$ Denote the j-th column of $\mathbf{W}_{\text{down}}^{(\ell)}$ as $\boldsymbol{w}_{\text{down}}^{(\ell),j} = [\mathbf{W}_{\text{down}}^{(\ell)}]_{:,j} \in \mathbb{R}^{d_h}$, which is the output vector of neuron (ℓ, j), and the neuron activation score at the i-th token position as $a_{i,j}^{(\ell)} = [\mathbf{a}^{(\ell)}]_{i,j}$.

2.2 MKE Problem Definition

Multimodal Knowledge Editing (MKE) aims to revise the factual knowledge stored in a pretrained multimodal model (mainly VLLMs) while preserving the related knowledge and general capabilities. Given a pretrained VLLM $f_{\mathcal{W}} : \mathcal{I} \times \mathcal{X} \to \mathcal{Y}$ parameterized by $\mathcal{W}$, we define an edit request as a triplet $\boldsymbol{e} = (\boldsymbol{I}_e, \boldsymbol{x}_e, \boldsymbol{y}_e)$, where $(\boldsymbol{I}_e, \boldsymbol{x}_e) \in \mathcal{I} \times \mathcal{X}$ is the image-text query pair and $\boldsymbol{y}_e \in \mathcal{Y}$ is the desired target output. The edit set is $\mathcal{D}_{\text{edit}} = \{\boldsymbol{e} \mid f_{\mathcal{W}}(\boldsymbol{I}_e, \boldsymbol{x}_e) \neq \boldsymbol{y}_e\}$. The target of MKE is to obtain an editing operator $\mathcal{E} : (\mathcal{W}, \boldsymbol{e}) \mapsto \mathcal{W}'$ that transforms $f_{\mathcal{W}}$ into $f_{\mathcal{W}'}$ such that

$$f_{\mathcal{W}'}(\boldsymbol{I}_e, \boldsymbol{x}_e) = \begin{cases} \boldsymbol{y}_e, & \text{if } (\boldsymbol{I}_e, \boldsymbol{x}_e, \boldsymbol{y}_e) \in \mathcal{D}_{\text{edit}}, \\ f_{\mathcal{W}}(\boldsymbol{I}_e, \boldsymbol{x}_e), & \text{otherwise.} \end{cases} \tag{2}$$

MKE can be evaluated by **Reliability**, textual and modality **Generality**, and textual and modal **Locality** (Cheng et al., 2023; Yao et al., 2023; Zhang et al., 2024). Note that when $\boldsymbol{I}_e \equiv \varnothing$, Eqn. 2 degenerates to pure-LLM knowledge editing.

3 Decoupled Vision–Text Localization

In this section, we illustrate how to localize knowledge-concerned neurons in VLLMs. We first hypothesize that multimodal factual knowledge is predominantly encoded within modality-aware neurons in FFNs (Dai et al., 2022; Geva et al., 2023; Pan et al., 2024a; Fang et al., 2024; Schwettmann et al., 2023; Liu et al., 2025; Li et al., 2026). We estimate and integrate modality-specific neuron attributions via decoupled vision–text activations to obtain candidate layers, in which we filter exceedingly high visually dominated neurons to identify the editing targets.

3.1 Modality-Specific Neuron Attributions

Denote the set of all neurons across all FFN layers in $f_{\mathcal{W}}$ as $\mathcal{N} = \{(\ell, j) \mid \ell \in [1, L],\, j \in [1, d_{\text{ffn}}]\}$, and the neurons for a specific layer ℓ as $\mathcal{N}_\ell$. We partition the total S token positions into the following intervals:

$$\mathcal{P}_{\text{vis}} = [1, N_v], \quad \mathcal{P}_{\text{targ}} \subseteq \mathcal{P}_{\text{txt}} = [N_v{+}1, S], \tag{3}$$

where $\mathcal{P}_{\text{vis}}$ is the vision-token range, $\mathcal{P}_{\text{targ}}$ is the subject-entity token range inside the prompt (the knowledge entity to be edited), and $\mathcal{P}_{\text{txt}}$ is the entire text block; $\mathcal{P}_{\text{all}} = \mathcal{P}_{\text{vis}} \cup \mathcal{P}_{\text{txt}} = [1, S]$. We capture the activation $a_{i,j}^{(\ell)}$ of neuron $(\ell, j) \in \mathcal{N}$ at the i-th token position. For any token position set $\mathcal{P} \subseteq \mathcal{P}_{\text{all}}$ and neuron $(\ell, j) \in \mathcal{N}$, we collect the max-min normalized modality activations across the d_{ffn} neurons and compute layer-wise modality-specific neuron attributions

$$\bar{a}_j^{(\ell),\mathcal{P}} = \frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} \hat{a}_{i,j}^{(\ell)},\ \bar{\boldsymbol{a}}^{(\ell),\mathcal{P}} = [\bar{a}_1^{(\ell),\mathcal{P}}, \ldots, \bar{a}_{d_{\text{ffn}}}^{(\ell),\mathcal{P}}]^\top, \tag{4}$$

where $\hat{a}_{i,j}^{(\ell)}$ denotes the normalized $a_{i,j}^{(\ell)}$. To quantify the contrastive coupling between vision and target activation, we compute their Pearson correlation coefficient:

$$\kappa_\ell = \frac{\big(\bar{\boldsymbol{a}}^{(\ell),\mathcal{P}_{\text{vis}}} - \tilde{a}^{(\ell),\mathcal{P}_{\text{vis}}}\mathbf{1}\big)^\top \big(\bar{\boldsymbol{a}}^{(\ell),\mathcal{P}_{\text{targ}}} - \tilde{a}^{(\ell),\mathcal{P}_{\text{targ}}}\mathbf{1}\big)}{\big\|\bar{\boldsymbol{a}}^{(\ell),\mathcal{P}_{\text{vis}}} - \tilde{a}^{(\ell),\mathcal{P}_{\text{vis}}}\mathbf{1}\big\|_2 \cdot \big\|\bar{\boldsymbol{a}}^{(\ell),\mathcal{P}_{\text{targ}}} - \tilde{a}^{(\ell),\mathcal{P}_{\text{targ}}}\mathbf{1}\big\|_2 + \epsilon}, \tag{5}$$

where $\mathbf{1} \in \mathbb{R}^{d_{\text{ffn}}}$, $\tilde{a}^{(\ell),\mathcal{P}} = \frac{1}{d_{\text{ffn}}} \sum_j \bar{a}_j^{(\ell),\mathcal{P}}$, and ϵ is the smoothing factor. κ_ℓ amplifies the score in layers where the vision and edit targets are globally coherent. To measure each neuron's contribution to the modality, we compute vision-text decoupled activation ratios, respectively,

$$\rho_j^{(\ell),\text{vis}} = \frac{\bar{a}_j^{(\ell),\mathcal{P}_{\text{vis}}}}{\left|\bar{a}_j^{(\ell),\mathcal{P}_{\text{all}}}\right| + \epsilon}, \qquad \rho_j^{(\ell),\text{targ}} = \frac{\bar{a}_j^{(\ell),\mathcal{P}_{\text{targ}}}}{\left|\bar{a}_j^{(\ell),\mathcal{P}_{\text{all}}}\right| + \epsilon}, \tag{6}$$

and fuse them as a unified score $s_{\ell,j} = \rho_j^{(\ell),\text{vis}} \cdot \rho_j^{(\ell),\text{targ}} \cdot (1 + |\kappa_\ell|)$ to capture the multimodal editing signals across layers.

3.2 Locating Knowledge-Concerned Neurons

Given $f_{\mathcal{W}}$, the goal of localization is to identify the candidate layer set $\mathcal{L} \subseteq [1, L]$ and target neurons $\mathcal{T}_\ell \subseteq \mathcal{N}_\ell\, (\ell \in \mathcal{L})$. We first execute layer-wise profiling selection and subsequently pick knowledge-concerned neurons via attribution scores and visual filters.

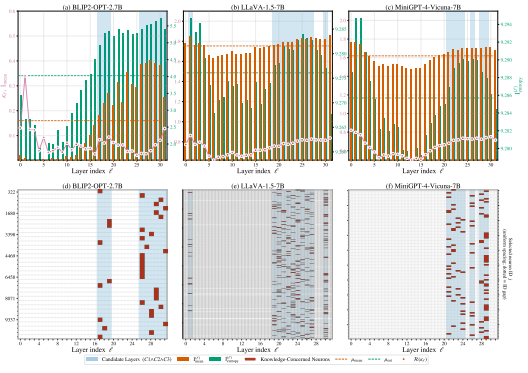

Figure 2: Localization Results. (a)-(c) visualize the tendency of I_{mean}, I_{entropy}, and κ_ℓ. (d)-(f) exhibit knowledge-concerned **neurons**. The candidate layers are in **bands**.

Layer-wise Profiling and Selection. To identify the candidate layer set $\mathcal{L}$, we compute $\rho_j^{(\ell),\text{vis}}, \rho_j^{(\ell),\text{targ}}, \kappa_\ell$ and $s_{\ell,j}$ across all L layers in a single forward pass and record κ_ℓ and two complementary statistics: $I_{\text{mean}}^{(\ell)} = \frac{1}{d_{\text{ffn}}} \sum_{j=1}^{d_{\text{ffn}}} |s_{\ell,j}|$ to measure the average engagement, and $I_{\text{entropy}}^{(\ell)} = -\sum_{j=1}^{d_{\text{ffn}}} p_{\ell,j} \log p_{\ell,j}$ to quantify how broadly the fused signal is distributed among neurons within the layer. Note that $p_{\ell,j} = |s_{\ell,j}| / \sum_k^{d_{\text{ffn}}} |s_{\ell,k}|$. Recent studies (Kaduri et al., 2025; Kang et al., 2025; Lin et al., 2025; Zhang et al., 2025) suggest that visual information leaks from vision tokens into the text residual stream as autoregressive depth grows, where a larger magnitude of κ_ℓ indicates a higher cross-modal information fusion. This indicates that the access to the multimodal knowledge distributed within the FFN is mediated by a sparse set of κ_ℓ rising phases. We therefore leverage the rising-phase of κ_ℓ and keep I_{mean} and I_{entropy} above the mean thresholds, which are not subject to this spike-inflation effect because they are computed over all d_{ffn} neurons rather than over a single cross-modal correlation coefficient. The resulting rule is architecture-agnostic: it admits any layer whose coupling is locally increasing (fusion is actively forming there, regardless of whether the absolute κ value sits above or below the model-wide average), and recovers a comparable candidate band across both regimes (Fig. 2, blue band). The candidate layer set $\mathcal{L}$ is defined as:

$$\mathcal{L} = \Big\{ \ell \;\Big|\; \underbrace{\ell \in \mathcal{R}(\kappa_\ell)}_{\text{(C1) Fusion}}, \underbrace{I_{\text{mean}}^{(\ell)} \geq \bar{I}_{\text{mean}}}_{\text{(C2) Engagement}}, \underbrace{I_{\text{entropy}}^{(\ell)} \geq \bar{I}_{\text{entropy}}}_{\text{(C3) Tracibility}} \Big\}, \tag{7}$$

where $\bar{I}_\star = \frac{1}{L} \sum_{\ell \in [1,L]} I_\star^{(\ell)}$ is the cross-layer average and $\mathcal{R}(\kappa_\ell) = \{\ell \mid \Delta\kappa(\ell) = \kappa_{\ell+1} - \kappa_\ell > 0,\ \ell \in [1, L-1]\}$ denotes the increasing interval. Each clause of the consensus $C1 \wedge C2 \wedge C3$ isolates a distinct property of layer ℓ: (i) $C1$ [$\ell \in \mathcal{R}(\kappa_\ell)$] reflects the cross-modal fusion as the visual signal is being re-injected into the residual stream; (ii) $C2$ [$\mathrm{I}_{\text{mean}}^{(\ell)} \geq \bar{I}_{\text{mean}}$] picks the involved FFN layers that substantively participate in the prediction; (iii) $C3$ [$\mathrm{I}_{\text{entropy}}^{(\ell)} \geq \bar{I}_{\text{entropy}}$] identifies the layers with higher activation entropy, which indicates a higher involvement of these layers in processing the current input. Specifically, high I_{entropy} indicates that the normalized activation distribution across neurons within layers are relatively uniform, where more neurons are simultaneously activated with comparable magnitudes. We further discuss this in Sec. 5.3 and Sec. 5.4.

Neuron-wise Filtering and Selection. Since visual tokens in the language space constitute a compressed representation of the underlying visual data manifold, any excessive modification of visual

neurons risks distorting the learned representation, which can lead to model collapse. Thus, we exclude neurons with visual activation over τ_{vis} from the filtering set

$$\mathcal{N}_\ell^{\text{filt}} = \big\{(\ell, j) \in \mathcal{N}_\ell \,\big|\, \rho_j^{(\ell),\text{vis}} < \tau_{\text{vis}}\big\}, \tag{8}$$

where $j \in [1, d_{\text{ffn}}]$ and $\tau_{\text{vis}} \in (0,\ 1{+}N_t/N_v)$ is a vision-dominance threshold ($N_v > 0$, see Appendix H for detailed estimation). We then pick the top-K_ℓ neurons from each layer $\ell \in \mathcal{L}$:

$$\mathcal{T}_\ell = \arg\max_{\mathcal{T} \subseteq \mathcal{N}_\ell^{\text{filt}}} \sum_{(\ell,j)\in\mathcal{T}} |s_{\ell,j}|, \text{ s.t. } |\mathcal{T}_\ell| = K_\ell. \tag{9}$$

The coupling multiplication $s_{\ell,j}$ measures the coordination among groups of visual and text target neurons within each layer, while the global top-K_ℓ neuron selection captures the cross-layer coherence. The full neuron-wise localization procedure is summarized in Algorithm 1.

4 Editing Knowledge-Concerned Neurons

Algorithm 1 DELTA – Localization

Require: $f_{\mathcal{W}}$, $\boldsymbol{e} = (\boldsymbol{I}_e, \boldsymbol{x}_e, \boldsymbol{y}_e)$, $\{K\}_{\ell\in\mathcal{L}}$, τ_{vis}.
Forward process on $(\boldsymbol{I}_e, \boldsymbol{x}_e, \boldsymbol{y}_e)$, recording $\hat{a}_{i,j}^{(\ell)}, \mathcal{T} \leftarrow \varnothing$.
for $\ell = 1, \ldots, L$ **do**
  Compute $\bar{\boldsymbol{a}}^{(\ell),\mathcal{P}}$ for $\mathcal{P} \in \{\mathcal{P}_{\text{vis}}, \mathcal{P}_{\text{targ}}, \mathcal{P}_{\text{all}}\}$
  Profile κ_ℓ, $s_{\ell,j}$, $\rho_j^{(\ell),\text{vis}}$, $\rho_j^{(\ell),\text{targ}}$
  Record $\{I_{\text{mean}}^{(\ell)}, I_{\text{entropy}}^{(\ell)}\}$
end for
Judge (C1)–(C3): $\mathcal{R}(\kappa_\ell)$, $\bar{I}_{\text{mean}}$, $\bar{I}_{\text{entropy}}$. Pick $\mathcal{L}$.
for $\ell \in \mathcal{L}$ **do**
  $\mathcal{N}_\ell^{\text{filt}} = \big\{(\ell, j) \in \mathcal{N}_\ell \,\big|\, \rho_j^{(\ell),\text{vis}} < \tau_{\text{vis}}\big\}$
  $\mathcal{T}_\ell \leftarrow \arg\text{top-}K_\ell \text{ of } \{|s_{\ell,j}|\}_j$
  $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{T}_\ell$
end for
return $\mathcal{T}$

We update the weights of knowledge-concerned neurons to edit the knowledge. Localization identifies the candidate layer set $\mathcal{L} \subseteq [1, L]$ and the target neurons $\mathcal{T} = \cup_{\ell\in\mathcal{L}}\mathcal{T}_\ell$. Given $\boldsymbol{e} = (\boldsymbol{I}_e, \boldsymbol{x}_e, \boldsymbol{y}_e)$, we optimize a per-neuron weight $\boldsymbol{\delta}_{\ell,j} \in \mathbb{R}^{d_h}$ for each $(\ell, j) \in \mathcal{T}_\ell \subseteq \mathcal{T}$, conditioned on $(\boldsymbol{I}_e, \boldsymbol{x}_e)$, while preserving the pre-edit model $f_{\mathcal{W}}$ on unrelated inputs. Concretely, our objective is the param shifts $\boldsymbol{\delta}_{\ell,j}$, which are added to the corresponding neurons of the ℓ-th layer FFN output (Eqn. 1) via:

$$\text{FFN}_{\boldsymbol{\delta}}^{(\ell)}\big(\tilde{\boldsymbol{h}}^{(\ell-1)}\big) = \mathbf{a}^{(\ell)}\, \mathbf{W}_{\text{down}}^{(\ell)\ \top} + \mathbf{a}_{:,\mathcal{T}_\ell}^{(\ell)}\, \boldsymbol{\Delta}^{(\ell)\top}, \tag{10}$$

where $\mathbf{a}_{:,\mathcal{T}_\ell}^{(\ell)} \in \mathbb{R}^{S\times|\mathcal{T}_\ell|}$ is the column-subset of $\mathbf{a}^{(\ell)}$ indexed by $\mathcal{T}_\ell$ and $\boldsymbol{\Delta}^{(\ell)} = \{\boldsymbol{\delta}_{\ell,j}\}_{(\ell,j)\in\mathcal{T}_\ell} \in \mathbb{R}^{d_h\times|\mathcal{T}_\ell|}$ stacks those param shifts at layer ℓ. For each $(\ell, j) \in \mathcal{T}_\ell$, the perturbation item $\mathbf{a}_{:,\mathcal{T}_\ell}^{(\ell)}\, \boldsymbol{\Delta}^{(\ell)\top}$ modifies the j-th column of $\mathbf{W}_{\text{down}}^{(\ell)}$ indexed by $\mathcal{T}_\ell$ via $\boldsymbol{w}_{\text{down}}^{(\ell),j} \leftarrow \boldsymbol{w}_{\text{down}}^{(\ell),j} + \boldsymbol{\delta}_{\ell,j}$, obtaining the post-edit weight $\mathcal{W}'$. We now illustrate how to obtain $\{\boldsymbol{\Delta}^{(\ell)}\}_{\ell\in\mathcal{L}}$ with the editing loss $\mathcal{L}_{\boldsymbol{\Delta}}$, consist of the following items.

4.1 Basic Components

For an image-text query $(\boldsymbol{I}_e, \boldsymbol{x}_e)$, denote the target output $\boldsymbol{y}_e = (\hat{y}_1, \ldots, \hat{y}_T)$ of length T. Following (Meng et al., 2022; 2023; Pan et al., 2025), we combine negative log-likelihood for reliable generation, a prefix-position KL divergence to prevent unrelated alterations, and a repetition penalty to avoid repetitive outputs, formally:

$$\begin{aligned}
\mathcal{L}_{\text{edit}} &= -\log P_{\mathcal{W}'}(\boldsymbol{y}_e \mid \boldsymbol{I}_e, \boldsymbol{x}_e), \\
\mathcal{L}_{\text{pref}} &= D_{\text{KL}}\big(\mathbb{P}_{\mathcal{W}'}(\cdot \mid \boldsymbol{I}_e, \boldsymbol{x}_e) \,\big\|\, \mathbb{P}_{\mathcal{W}}(\cdot \mid \boldsymbol{I}_e, \boldsymbol{x}_e)\big), \\
\mathcal{L}_{\text{rep}} &= -\log\big(1 - P_{\mathcal{W}'}(\hat{y}_T \mid \boldsymbol{I}_e, \boldsymbol{x}_e, \boldsymbol{y}_e)\big),
\end{aligned} \tag{11}$$

where $\mathbb{P}_{\mathcal{W}}(\cdot \mid \boldsymbol{I}_e, \boldsymbol{x}_e)$ is the position-wise predicted-token distribution of $f_{\mathcal{W}}$, except at the target position for $\boldsymbol{y}_e$. Note that the autoregressive sequence probability $P_{\mathcal{W}}(\boldsymbol{y}_e \mid \boldsymbol{I}_e, \boldsymbol{x}_e) = \prod_{t=1}^{T} P_{\mathcal{W}}(\hat{y}_t \mid \boldsymbol{I}_e, \boldsymbol{x}_e, \hat{y}_{<t})$.

4.2 Visual Perturbation

Since VLLMs are likely to be sensitive to perturbations at visual tokens in the embedding space (Jeong et al., 2026; Wang et al., 2026; Seo et al., 2026), while visual variation (e.g., perspective, color temperature, etc.) is frequently encountered in practice, the null-space regularization

may constrain $\{\mathbf{\Delta}^{(\ell)}\}_{\ell\in\mathcal{L}}$ to latch onto subspaces that merely correlate with $\boldsymbol{I}_e$ and lead to generality degradation. Thus, we inject stochastic noise into the initial embeddings at visual-token positions via $\widetilde{\boldsymbol{z}}_{\text{vis}} = \boldsymbol{z}_{\text{vis}} + \boldsymbol{\xi}$, where $\boldsymbol{\xi} \sim \mathcal{N}\big(\mathbf{0},\, \text{var}[\eta_{\text{vis}} \cdot \boldsymbol{z}_{\text{vis}}] \cdot \mathbf{I}_{N_v \times d_h}\big)$, $\text{var}(\cdot)$ denotes the elemental variance in $\boldsymbol{z}_{\text{vis}}$. We priorly compute the target-position logits $\widetilde{\boldsymbol{y}} \in \mathbb{R}^{T\times V}$ via $f_{\mathcal{W}}$ with the perturbed initial hidden state as $\widetilde{\boldsymbol{h}}^{(0)} = [\widetilde{\boldsymbol{z}}_{\text{vis}}; \boldsymbol{z}_{\text{txt}}]$. During optimization, we collect the target-position logits $\boldsymbol{y}^* \in \mathbb{R}^{T\times V}$ with the same perturbed initial $\widetilde{\boldsymbol{h}}^{(0)}$ and then regularize the edit through

$$\mathcal{L}_{\text{vis}} = D_{\text{KL}}\big(\,\text{softmax}(\boldsymbol{y}^*)\;\big\|\;\text{softmax}(\widetilde{\boldsymbol{y}})\,\big), \tag{12}$$

where $f_{\mathcal{W}}$ acts as a stability regularizer on $\mathbf{\Delta}^{(\ell)}$ via $\mathcal{L}_{\text{vis}}$ and discourages $\mathbf{\Delta}^{(\ell)}$ from overfitting to the exact token-level statistics of $\boldsymbol{I}_e$, complementing the null-space projection in confining $\mathbf{\Delta}^{(\ell)}$ to a stable, semantically grounded subspace.

4.3 Null-Space Regularized Optimization

Denote the column sub-matrix at the ℓ-th layer by $\mathbf{A}^{(\ell)} = \mathbf{a}^{(\ell)}_{:,\mathcal{T}_\ell} \in \mathbb{R}^{S\times|\mathcal{T}_\ell|}$, where $|\mathcal{T}_\ell| \ll S$. Inspired by Adam-NSCL (Wang et al., 2021) and AlphaEdit (Fang et al., 2025), we optimize $\big\|\mathbf{A}^{(\ell)}\mathbf{\Delta}^{(\ell)^\top}\big\|_F^2 = \text{tr}(\mathbf{\Delta}^{(\ell)}\mathbf{A}^{(\ell)^\top}\mathbf{A}^{(\ell)}\,\mathbf{\Delta}^{(\ell)^\top})$ to preserve the unrelated knowledge within $f_{\mathcal{W}'}$. Let $\mathbf{A}^{(\ell)} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ be the SVD, we collect the smallest d_{null} singular values corresponding to d_{null} right singular vectors within $\{\mathbf{v}_i^{(\ell)}\}_{i=1}^{|\mathcal{T}_\ell|}$ ordered by ascending eigenvalue, defined as $\mathbf{V}_\perp^{(\ell)} = \Big[\mathbf{v}_1^{(\ell)}, \ldots, \mathbf{v}_{d_{\text{null}}}^{(\ell)}\Big] \in \mathbb{R}^{|\mathcal{T}_\ell|\times d_{\text{null}}}$. The projection matrix onto the approximate null space of $\mathbf{A}^{(\ell)}$ is thus $\mathbf{P}^{(\ell)}_{d_{\text{null}}} = \mathbf{V}_\perp^{(\ell)}\mathbf{V}_\perp^{(\ell)^\top}$. Considering projecting $\mathbf{\Delta}^{(\ell)^\top}$ onto the null-space of $\mathbf{A}^{(\ell)}$ at every optimization step can aggravate the gradient flow between the editing target and the discarded row-space component, which potentially destabilizes the convergence. We therefore penalize the row-space component of $\mathbf{\Delta}^{(\ell)^\top}$, formulated as:

$$\mathcal{L}_{\text{ns}} = \sum_{\ell\in\mathcal{L}} \big\|(\mathbf{I} - \mathbf{P}^{(\ell)}_{d_{\text{null}}})\,\mathbf{\Delta}^{(\ell)^\top}\big\|_F^2, \tag{13}$$

where $\mathbf{I} - \mathbf{P}^{(\ell)}_{d_{\text{null}}}$ projects vectors onto the row-space directions where the preserved inputs actively excite. We provide detailed derivation and theoretical analysis in Appendix C.

Optimization and Hard Projection. All objectives are weighted by $\alpha_{\text{pref}}, \alpha_{\text{rep}}, \lambda_{\text{ns}}$, and λ_{vis} , organized as

$$\mathcal{L}_{\mathbf{\Delta}} = \mathcal{L}_{\text{edit}} + \alpha_{\text{pref}} \cdot \mathcal{L}_{\text{pref}} + \alpha_{\text{rep}} \cdot \mathcal{L}_{\text{rep}} + \lambda_{\text{ns}} \cdot \mathcal{L}_{\text{ns}} + \lambda_{\text{vis}} \cdot \mathcal{L}_{\text{vis}}. \tag{14}$$

We apply the hard projection as a one-time correction step via $\mathbf{\Delta}^{(\ell)} \leftarrow \mathbf{\Delta}^{(\ell)}\mathbf{P}^{(\ell)}_{d_{\text{null}}}$, which constrains the perturbation to lie in $\text{Col}(\mathbf{V}_\perp^{(\ell)})$ and to remove residual row-space components.

5 Experiments

5.1 Implementation Details

VLLM Backbones & Baselines. Experiments are mainly conducted on BLIP2-OPT-2.7B (Li et al., 2023), LLaVA-v1.5-7B (Liu et al., 2023), and MiniGPT4-Vicuna-7B (Zhu et al., 2024). We compare **DELTA** with the following baselines. *Training-based* methods relies on external memory modules, additional parameters, or the training required auxiliary editors: SERAC (Mitchell et al., 2022b), MEND (Mitchell et al., 2022a), UniKE (Pan et al., 2024b), VisEdit (Chen et al., 2025b), LiveEdit (Chen et al., 2025a), ODEdit (Su et al., 2026). *Locate-then-edit* is previous developped only for pure-LLMs and mainly relies on causal training: ROME (Meng et al., 2022), MEMIT (Meng et al., 2023), PMET (Li et al., 2024), MLE (Zhou et al., 2025) (edit multi-granularity multimodal knowledge), FiNE (Pan et al., 2025), AlphaEdit (Fang et al., 2025). We reproduce them to VLLMs for evaluation.

Benchmark & Metrics. We conduct experiments on Editing VQA (E-VQA) and Editing Image Captioning (E-IC) from MMEdit benchmark (Cheng et al., 2023), and VLKEB (Huang et al., 2024) on **Reliability**, **Generality** (modality and text), and **Locality** (modality and text).

Table 1: Single-edit performance on BLIP2, LLaVA1.5, and MiniGPT4 across 3 datasets. The best and second in each column are in **bold** and underline. *OOD* rows reuse the EVQA-trained weights to perform out-of-distribution evaluation.

Method	EIC						EVQA						VLKEB					
	Rel.	T-Gen.	M-Gen.	T-Loc.	M-Loc.	*Avg.*	Rel.	T-Gen.	M-Gen.	T-Loc.	M-Loc.	*Avg.*	Rel.	T-Gen.	M-Gen.	T-Loc.	M-Loc.	*Avg.*
BLIP2-OPT (2.7B)																		
SERAC	29.41	30.98	29.11	98.39	97.16	57.01 (±0.30)	92.78	87.77	91.13	92.65	67.20	86.31 (±0.12)	36.42	38.33	36.09	64.07	18.22	38.63 (±0.32)
MEND	28.85	28.04	28.87	1.93	0.74	17.69 (±0.71)	32.52	31.13	29.38	1.02	0.56	18.92 (±0.44)	38.84	38.34	38.79	1.14	0.75	23.57 (±0.51)
ODEdit(+MEND)	11.63	11.71	11.79	0.00	0.00	7.03 (±0.28)	85.52	84.99	85.20	0.79	0.33	51.37 (±0.36)	26.43	25.18	26.18	3.46	1.58	16.57 (±0.23)
UniKE	79.98	73.04	79.88	2.28	1.50	47.34 (±0.56)	94.21	94.12	94.21	6.24	0.20	57.80 (±0.48)	93.67	91.30	93.42	1.31	1.50	56.24 (±0.41)
ROME	58.48	77.93	44.84	99.51	97.74	75.70 (±0.06)	29.23	30.69	28.81	97.31	77.23	52.65 (±0.01)	6.71	6.52	6.66	43.44	99.18	32.50 (±0.02)
MEMIT	29.83	31.10	29.42	93.63	91.27	55.05 (±0.41)	1.87	1.92	1.82	98.48	98.10	40.44 (±0.21)	6.04	6.00	6.07	76.45	97.41	38.39 (±0.08)
MLE	23.26	21.51	22.28	34.92	5.50	21.50 (±0.14)	9.73	8.50	8.78	48.44	8.60	16.81 (±0.11)	19.82	20.59	19.99	54.65	19.12	26.84 (±0.09)
PMET	30.84	31.69	30.69	88.71	89.37	54.26 (±0.28)	1.86	1.92	1.80	95.73	94.84	39.23 (±0.68)	6.12	5.95	6.16	92.42	90.55	40.24 (±0.31)
AlphaEdit	29.63	31.10	29.34	99.94	98.48	57.70 (±0.14)	1.92	1.90	1.79	99.54	99.85	41.00 (±0.12)	6.02	5.97	6.06	99.45	99.37	43.37 (±0.06)
FiNE	**100.00**	89.47	88.26	80.95	39.17	79.17 (±0.15)	**100.00**	90.90	90.96	67.88	15.97	73.14 (±0.24)	100.00	94.86	92.93	45.11	21.47	70.87 (±0.21)
VisEdit	48.47	48.83	46.95	**100.00**	85.88	66.03 (±0.89)	84.31	83.56	82.87	**100.00**	61.85	82.52 (±0.60)	73.83	73.59	73.44	**100.00**	65.44	77.26 (±0.94)
VisEdit(OOD)	13.25	13.89	13.04	100.00	42.40	36.52 (±0.91)	–	–	–	–	–	–	8.06	8.18	8.33	100.00	52.99	35.51 (±0.98)
LiveEdit	93.46	**94.57**	93.22	100.00	**100.00**	**96.25** (±1.40)	92.17	**93.02**	84.45	100.00	**100.00**	**93.93** (±1.42)	96.96	**96.81**	95.23	100.00	**100.00**	**97.80** (±1.74)
LiveEdit(OOD)	32.10	31.69	29.72	100.00	68.73	52.45 (±1.48)	–	–	–	–	–	–	27.95	30.05	29.68	100.00	69.52	51.44 (±1.59)
DELTA(Ours)	97.37	83.71	**95.11**	96.78	89.51	92.50 (±0.17)	96.28	85.48	**94.91**	99.58	83.96	92.04 (±0.13)	**100.00**	96.08	**99.95**	94.66	87.80	95.70 (±0.09)
LLaVA-1.5 (7B)																		
SERAC	41.50	44.91	44.90	89.84	28.49	49.93 (±0.39)	81.13	80.21	81.82	93.24	55.03	78.29 (±0.45)	88.45	86.64	87.20	68.19	21.25	70.35 (±0.59)
MEND	88.87	79.43	81.35	81.05	81.78	82.50 (±0.61)	85.65	75.94	83.12	82.10	83.49	82.06 (±1.07)	78.42	65.78	65.04	70.13	68.18	69.51 (±0.44)
ODEdit(+MEND)	29.13	29.34	33.32	72.54	71.40	47.15 (±0.86)	42.26	39.57	42.13	10.74	13.70	29.68 (±0.77)	24.43	23.66	23.77	7.68	4.08	16.72 (±0.74)
UniKE	30.20	28.72	26.50	15.78	13.83	23.01 (±0.53)	77.20	72.29	73.21	13.99	8.67	49.07 (±0.60)	62.60	55.37	61.56	12.64	10.71	40.58 (±0.61)
ROME	56.89	53.27	53.79	89.53	89.13	68.52 (±0.23)	88.75	77.35	85.77	89.59	76.95	83.68 (±0.30)	80.23	81.36	80.75	85.19	88.89	83.28 (±0.17)
MEMIT	51.34	51.65	48.34	87.21	85.28	64.76 (±0.84)	69.88	60.25	63.18	89.40	93.07	75.16 (±0.68)	50.53	52.19	50.59	87.33	87.25	65.58 (±0.54)
MLE	1.16	1.94	1.21	78.20	0.36	16.44 (±0.17)	3.74	6.24	5.87	96.63	1.20	22.04 (±0.18)	2.92	3.75	2.58	81.87	1.23	18.14 (±0.15)
PMET	52.84	53.53	49.18	94.86	93.14	68.71 (±0.33)	32.97	30.81	28.67	92.96	92.65	55.61 (±0.28)	47.18	50.07	49.82	94.01	93.18	66.85 (±0.27)
AlphaEdit	52.83	53.56	49.09	**100.00**	**100.00**	71.10 (±0.15)	67.79	64.66	64.84	97.83	89.13	76.85 (±0.14)	60.67	61.09	60.54	85.19	93.33	72.16 (±0.09)
FiNE	100.00	52.94	49.10	100.00	52.80	70.97 (±0.19)	100.00	29.89	26.64	100.00	31.67	57.64 (±0.27)	100.00	75.15	79.57	92.84	89.69	87.45 (±0.22)
VisEdit	51.14	50.41	48.09	100.00	66.48	63.22 (±1.47)	93.97	**92.13**	**92.31**	**100.00**	87.33	93.15 (±1.08)	59.27	61.46	59.37	**100.00**	83.02	72.62 (±1.52)
VisEdit(OOD)	38.98	40.16	38.82	100.00	50.11	53.61 (±0.98)	–	–	–	–	–	–	36.28	44.87	41.08	100.00	49.92	54.43 (±1.09)
LiveEdit	90.31	**91.97**	**91.42**	100.00	100.00	94.74 (±0.45)	94.01	92.02	89.08	100.00	**100.00**	**95.02** (±0.52)	84.94	**87.73**	79.61	100.00	**100.00**	90.46 (±0.87)
LiveEdit(OOD)	75.55	78.50	72.25	100.00	77.04	80.67 (±0.70)	–	–	–	–	–	–	69.88	77.36	73.95	100.00	69.98	78.23 (±0.68)
DELTA(Ours)	**99.39**	91.93	89.05	97.88	96.25	**94.90** (±0.21)	**100.00**	89.88	89.85	98.87	94.83	94.69 (±0.16)	**100.00**	86.18	**98.04**	97.86	91.60	**94.74** (±0.12)
MiniGPT-4																		
SERAC	41.06	34.99	36.74	86.23	21.03	44.01 (±0.87)	58.12	62.60	62.97	86.10	29.24	59.81 (±0.59)	41.81	44.38	41.57	80.41	23.90	46.41 (±1.22)
MEND	75.81	64.79	45.13	74.82	59.38	63.99 (±0.76)	37.51	32.16	29.34	49.96	28.60	35.51 (±0.54)	9.81	9.29	9.37	0.98	0.92	6.07 (±1.40)
ODEdit(+MEND)	13.63	14.24	12.86	17.95	10.88	13.91 (±0.47)	61.09	58.97	63.68	37.60	29.62	50.19 (±0.73)	65.10	64.75	63.69	40.52	31.94	53.20 (±0.62)
UniKE	29.02	27.79	26.12	13.54	12.97	21.89 (±1.04)	79.86	79.53	77.51	10.93	9.42	51.45 (±0.92)	63.59	63.00	62.44	7.50	7.49	40.80 (±0.94)
ROME	49.75	48.29	47.13	87.68	88.28	64.23 (±0.18)	82.02	73.20	78.10	89.25	74.22	79.36 (±0.21)	70.31	73.23	71.50	84.09	89.31	77.69 (±0.09)
MEMIT	47.63	46.86	44.53	92.48	90.03	64.30 (±0.18)	48.49	35.78	48.73	92.58	92.29	63.57 (±0.33)	44.45	45.26	43.58	93.51	93.49	64.06 (±0.20)
MLE	38.66	34.19	38.14	83.27	38.82	46.62 (±0.18)	28.64	31.97	28.37	88.20	31.07	41.65 (±0.23)	62.33	58.12	60.37	81.72	47.50	62.01 (±0.21)
PMET	48.63	47.79	46.04	92.04	91.83	65.27 (±0.16)	33.76	31.83	37.30	94.01	93.98	58.18 (±0.21)	44.33	45.21	43.29	89.79	89.83	62.49 (±0.09)
AlphaEdit	48.60	47.80	46.08	98.68	97.92	67.82 (±0.26)	67.83	59.74	65.52	96.37	90.48	75.99 (±0.17)	46.19	46.76	45.95	94.43	93.59	65.38 (±0.15)
FiNE	100.00	80.93	**87.17**	78.28	49.59	79.19 (±0.19)	**100.00**	87.59	94.55	91.91	53.69	85.55 (±0.14)	98.80	98.74	98.75	72.47	55.67	84.89 (±0.26)
VisEdit	81.48	81.05	80.18	**100.00**	95.94	87.73 (±0.84)	85.70	83.58	84.52	**100.00**	89.75	88.71 (±0.73)	55.80	57.68	55.81	**100.00**	83.06	70.47 (±1.30)
VisEdit(OOD)	33.04	34.61	33.46	100.00	49.36	50.09 (±0.87)	–	–	–	–	–	–	36.58	40.37	38.01	100.00	52.49	53.49 (±0.60)
LiveEdit	87.24	**89.95**	82.35	100.00	**100.00**	**91.91** (±0.29)	86.92	81.30	89.13	100.00	**100.00**	91.47 (±0.45)	86.88	88.04	72.12	100.00	**100.00**	89.41 (±1.01)
LiveEdit(OOD)	76.88	78.04	62.12	100.00	76.92	78.79 (±0.65)	–	–	–	–	–	–	70.74	63.05	60.97	100.00	69.95	72.94 (±0.71)
DELTA(Ours)	**91.10**	84.11	80.46	98.70	89.25	88.72 (±0.75)	99.94	**91.74**	**98.99**	98.81	85.99	**95.09** (±0.18)	**100.00**	**99.64**	**99.77**	97.88	85.39	**96.54** (±0.24)

Settings. We set hyper-parameter tuple $(\alpha_{\text{pref}}, \alpha_{\text{rep}}, \lambda_{\text{ns}}, \lambda_{\text{vis}}) = (1.0, 0.01, 1.0, 0.7)$ and smoothing factor $\epsilon = 10^{-8}$ for each candidate layer. We provide detailed settings in Appendix D. All experiments are completed on 3 NVIDIA RTX A6000 GPUs and 2 NVIDIA RTX 3090 GPUs.

Table 2: Ablations by removing constraints.

	Rel	T-Gen	M-Gen	T-Loc	M-Loc	*Avg*
DELTA (Full)	**100.00**	96.08	99.95	94.66	**87.80**	**95.70**
w/o $\mathcal{L}_{\text{ns}}$	100.00	96.09 $^{+0.01}$	99.75 $^{-0.20}$	93.62 $^{-0.94}$	80.49 $^{-7.31}$	93.99 $^{-1.38}$
w/o $\mathcal{L}_{\text{pref}}$	100.00	96.33 $^{+0.25}$	**100.00** $^{+0.05}$	**94.75** $^{+0.09}$	81.24 $^{-6.56}$	94.46 $^{-1.24}$
w/o $\mathcal{L}_{\text{rep}}$	100.00	96.14 $^{+0.06}$	99.75 $^{-0.20}$	93.85 $^{-0.81}$	80.52 $^{-7.28}$	94.05 $^{-1.65}$
w/o $\mathcal{L}_{\text{vis}}$	100.00	**97.22** $^{+1.14}$	99.80 $^{-0.15}$	93.09 $^{-1.57}$	80.93 $^{-6.87}$	94.21 $^{-1.49}$

5.2 General Editing Performance

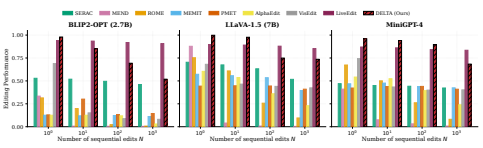

Figure 3: Sequential edit average performance.

Tab. 1 displays the single-edit results. We observe that previous locate-then-edit methods like ROME/MEMIT/PMET/AlphaEdit achieve average performance of about 60%–70% and always achieve high locality but low reliability (e.g., ROME/MEMIT/AlphaEdit), which indicates editing failures. The training-based parameter-expansion methods (e.g., VisEdit/LiveEdit) generally achieve high locality scores and perform better, while their editing performance is considerably more sensitive to the underlying data distribution. However, we observe that DELTA performs slightly worse than LiveEdit on the small-scale VLLM (e.g., BLIP2). We suggest that this discrepancy stems from the higher parameter density of smaller-scale VLLMs, which may make precise neuron selection more challenging. In contrast, the training process allows the auxiliary editor to better adapt to the internal signal strengths and activation patterns. This hypothesis also provides a potential explanation for the OOD degradation observed in training-based methods. For example,

when the editor weights trained on the EVQA dataset are directly applied to EIC (OOD evaluation), both LiveEdit and VisEdit exhibit a substantial drop in editing performance on BLIP2, while the magnitude of this degradation decreases as the model scale increases (LLaVA/MiniGPT4). These results suggest that the effectiveness of a trained editor may depend more strongly on its adaptation to model- and data-specific signal distributions, particularly in smaller-scale VLLMs. From this perspective, our **DELTA** is more suitable for general knowledge editing with consistently strong editing performance across different datasets without requiring the training of an auxiliary editor. We also compare the reliability of multimodal knowledge editing approaches in sequential edit (lifelong knowledge editing) across EIC, EVQA, and VLKEB datasets in Fig. 3. While LiveEdit is designed for lifelong editing with knowledge restored within the mixture-of-experts, **DELTA** still outperforms on 1-edit (BLIP2), 1/10-edits (LLaVA), and 1/10/100-edits (MiniGPT4) cases. All results above demonstrate the superiority of our **DELTA** method. Due to space limitations, we provide more results on the latest VLLMs (e.g., Qwen3 (Bai et al., 2025)) in Appendix E.

5.3 Ablation Study

Removing editing constraints. We remove constraints and assess a single edit with BLIP2-OPT-2.7B on VLKEB. As shown in Tab. 2, we observe that reliability is saturated at 100.00, while a trade-off appears among T-Gen/M-Gen and T-Loc/M-Loc. Both null-space regularization and visual perturbation protect the locality without interfering with the reliability. Note that for "w/o $\mathcal{L}_{\text{ns}}$" and "w/o $\mathcal{L}_{\text{vis}}$" cases, the projection and perturbation are entirely removed.

Recombination of diverse strategies. With recombination of typical *locate-then-edit* approaches on EVQA dataset, as shown in Fig. 4, we find that **DELTA** significantly outperforms other methods and combinations, as evidenced by the following observation: **1)** With the same editing method, locating with DELTA is generally the best; **2)** With the same localization method, editing with DELTA is generally the best; **3)** Results related to DELTA show obviously higher stability for long-term editing.

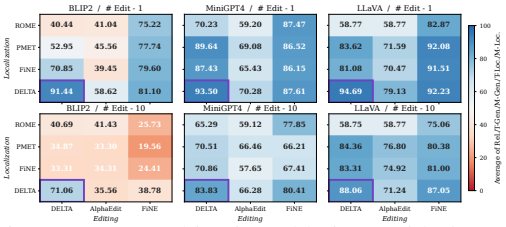

Figure 4: Recombination ablations with *locate-then-edit* methods. The horizontal and vertical axes enumerate the editing and locating strategies.

Varying picked layers. To verify the effect of conditions (C1)–(C3) for the selection of candidate layers, we performed ablations over all 2^3 combinations *(a)*–*(g)* and single-layer editing on BLIP2-OPT-2.7B with the EVQA dataset. As shown in Tab. 3, omitting any criterion induces a consistent average performance drop. Other single-layer cases performing extremely high *T/M-Loc* while poor *Rel* indicate editing failures. Note that case *(g)* corresponds to the all-layer continuous editing (no conditions), it is reasonable that knowledge is fully injected to the backbone and gains the highest *Rel*, however, destroys the re-injection of visual information flow and gains the worst *M-Loc*. This corresponds to our initial hypothesis. Detailed combinations and results on each combination are deferred to Appendix I. We also observe similar phenomena in sequential editing scenarios.

Table 3: Layer ablation on the EVQA dataset.

Cond	Rel	T-Gen	M-Gen	T-Loc	M-Loc	Avg
(a)	92.06 (−4.38%)	74.96 (−12.31%)	89.52 (−5.68%)	98.59 (−0.99%)	79.30 (−5.55%)	86.88 (−5.61%)
(b)	93.84 (−2.53%)	84.09 (−1.63%)	93.58 (−1.40%)	99.41 (−0.17%)	74.43 (−11.35%)	89.07 (−3.23%)
(c)	94.50 (−1.85%)	83.75 (−2.02%)	91.14 (−3.97%)	99.37 (−0.21%)	76.29 (−9.13%)	89.01 (−3.29%)
(d)	90.59 (−5.91%)	78.83 (−7.78%)	89.71 (−5.48%)	94.06 (−5.54%)	73.05 (−12.99%)	85.25 (−7.38%)
(e)	92.32 (−4.11%)	76.37 (−10.66%)	91.46 (−3.64%)	98.44 (−1.14%)	73.05 (−12.99%)	86.33 (−6.20%)
(f)	95.51 (−0.80%)	80.56 (−5.76%)	93.24 (−1.76%)	98.74 (−0.84%)	79.83 (−4.92%)	89.58 (−2.67%)
(g)	**97.98** (+1.77%)	84.12 (−1.59%)	93.33 (−1.67%)	99.46 (−0.12%)	51.88 (−38.21%)	85.35 (−7.27%)
17	17.13 (−82.21%)	8.95 (−89.53%)	16.17 (−83.00%)	99.68 (+0.10%)	98.90 (+17.79%)	48.17 (−47.66%)
18	17.19 (−82.15%)	9.01 (−89.46%)	16.44 (−82.72%)	99.41 (−0.17%)	98.43 (+17.23%)	48.10 (−47.74%)
19	17.17 (−82.17%)	9.51 (−88.87%)	15.82 (−83.37%)	99.37 (−0.21%)	97.29 (+15.88%)	47.83 (−48.03%)
26	17.08 (−82.26%)	9.31 (−89.11%)	16.07 (−83.10%)	**99.85** (+0.27%)	99.59 (+18.62%)	48.38 (−47.44%)
27	17.10 (−82.24%)	8.75 (−89.76%)	16.47 (−82.68%)	99.84 (+0.26%)	99.50 (+18.51%)	48.33 (−47.49%)
28	17.18 (−82.16%)	9.48 (−88.91%)	16.37 (−82.79%)	99.61 (+0.03%)	99.57 (+18.59%)	48.45 (−47.26%)
29	17.14 (−82.20%)	8.71 (−89.81%)	16.07 (−83.10%)	99.77 (+0.19%)	**99.86** (+18.94%)	48.31 (−47.51%)
30	17.11 (−82.23%)	9.01 (−89.46%)	16.24 (−82.93%)	99.80 (+0.22%)	99.80 (+18.87%)	48.39 (−47.43%)
31	17.16 (−82.18%)	9.15 (−89.30%)	16.24 (−82.93%)	99.77 (+0.19%)	99.29 (+18.26%)	48.32 (−47.50%)
Full	96.28	**85.48**	**94.91**	99.58	83.96	**92.04**

Impact of parameters. We explore the impact of vision-dominance threshold τ_{vis}, visual perturbation noise scale η_{vis}, and the null-space regularization dimension d_{null}, as exhibited in Fig. 5 and Fig. 6 on EVQA dataset. We observe: **1)** The average editing performance is not sensitive to visual perturbation noise scale η_{vis} but cannot be eliminated (see Tab. 2). **2)** The vision-dominance threshold τ_{vis} should be tuned carefully in real-scene knowledge editing practice, as it mainly affects the overall performance by influenc-

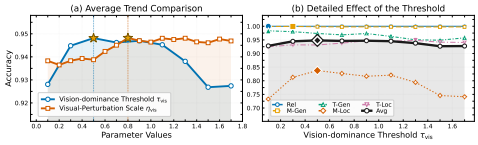

Figure 5: Parameter ablations. (a) shows the variation trend of τ_{vis} and η_{vis}; (b) shows τ_{vis} details.

ing modal-locality (M-Loc). **3)** increasing dimension d_{null} of the approximate null-space declines M-Loc/T-Loc while the others (Rel/M-Gen/T-Gen) are improved, revealing a trade-off among reliability, generality, and locality, which matches the latest theoretical evidence (Wang et al., 2024).

5.4 Localization Discussions

Localization Discussions. Fig. 2 illustrates the variation trends of $\kappa_\ell, I_{\text{mean}}$, and I_{entropy} across layers, alongside a visualization of the localized knowledge-concerned neurons. By analyzing over 1000 editing samples across different VLLM backbones, we observe a cross-model consensus: the three criteria (C1)-(C3) consistently highlight specific intermediate layers, rather than uniformly distributing across the entire network (Appendix G). In practice, the resulting layer sets are nearly invariant to the choice of edit instances within a backbone, so $\mathcal{L}$ can be fixed once per backbone as an offline profiling step, whereas the neuron-level localization ($s_{\ell,j}$ and τ_{vis} filtering) remains instance-dependent and is executed for each edit request. Notably, this pattern diverges fundamentally from the architectural priors adopted in locate-then-edit methods mainly based on causal tracing: ROME (Meng et al., 2022) constrains edits to a single fixed MLP layer, while MEMIT (Meng et al., 2023) operates on contiguous intermediate layers. We hypothesize that multimodal knowledge storage is structurally decoupled, where visual evidence flow (Zhang et al., 2025) is intermittently re-injected into the textual residual stream.

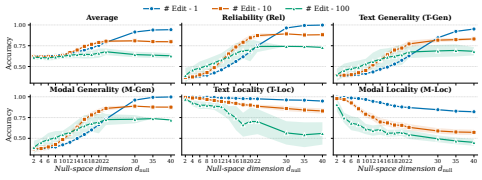

Figure 6: Parameter ablation with d_{null}. We perform sequential editing with BLIP2-OPT-2.7B .

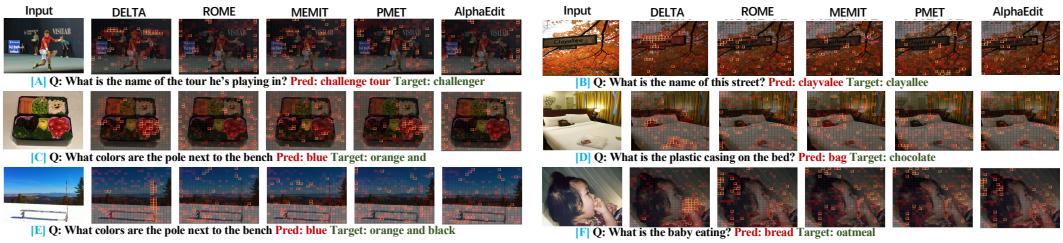

Figure 7: Visual evidence visualized with LLaVA. See Appendix F for more results.

5.5 Instance Analysis

To further quantify the visual evidence that each *locate-then-edit* method actually relies on, we record all neurons selected by corresponding localization methods and subsequently investigate which visual tokens contribute to them. Note that for layer-wise localization methods that only specify a layer band without per-neuron selection, we pick neurons with top-K (the total number of edited neurons in DELTA for fairness) activations. As shown in Fig. 7, *(A)(B)* cases exhibit that only DELTA localizes the neurons that focus on target characters; *(C)(D)* cases show that DELTA is also capable of locating tiny objects. These suggest that the decoupled visual and textual branches during localization successfully concentrate on the semantic objects corresponding to the targets. In contrast, other baselines yield either diffuse activations or heatmaps dominated by background regions. These observations suggest that explicit cross–modal localization is the missing ingredient in prior localization approaches, while DELTA bridges this gap.

6 Conclusion

We propose **DELTA**(Δ): a decoupled vision-text localization and editing framework via neuron attribution for knowledge editing in vision large language models. The framework is the first fine-grained neuron-level *locate-then-edit* approach specifically proposed for multimodal knowledge editing. Extensive experiments and analysis corroborate the effectiveness and superiority of each stage, paving the way for general-purpose knowledge editing for vision large language models.

AI USE STATEMENT

In this paper, we used generative AI tools only to improve the clarity, grammar, and presentation of the manuscript. The authors have carefully checked every AI-generated piece of content. All research ideas, mathematical formulas, and experimental results are all verified and checked by the authors, who take full responsibility for the paper.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Qizhou Chen, Chengyu Wang, Dakan Wang, Taolin Zhang, Wangyue Li, and Xiaofeng He. Lifelong knowledge editing for vision language models with low-rank mixture-of-experts. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 9455–9466, 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 Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13877–13888, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.856. URL https://aclanthology.org/2023.emnlp-main.856/.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8493–8502, 2022.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6491–6506, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.522. URL https://aclanthology.org/2021.emnlp-main.522/.

Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. Calibrating factual knowledge in pretrained language models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 5937–5947, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.438. URL https://aclanthology.org/2022.findings-emnlp.438/.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Junfeng Fang, Zongze Bi, Ruipeng Wang, Houcheng Jiang, Yuan Gao, Kun Wang, An Zhang, Jie Shi, Xiang Wang, and Tat-Seng Chua. Towards neuron attributions in multi-modal large language models. *Advances in Neural Information Processing Systems*, 37:122867–122890, 2024.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. Alphaedit: Null-space constrained model editing for language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=HvSytvg3Jh.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12216–12235, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.751. URL https://aclanthology.org/2023.emnlp-main.751/.

Wenyi Hong, Xiaotao Gu, Ziyang Pan, Zhen Yang, Yuting Wang, Yue Wang, Yuanchang Yue, Yu Wang, Yanling Wang, Yan Wang, et al. Glm-5v-turbo: Toward a native foundation model for multimodal agents. *arXiv preprint arXiv:2604.26752*, 2026.

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.

Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. Transformer-patcher: One mistake worth one neuron. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=4oYUGeGBPm.

Sungheon Jeong, Yoojeong Song, and Hyungjoon Kim. Understanding the visual projection space of multimodal llms. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 6049–6058, 2026.

Houcheng Jiang, Junfeng Fang, Tianyu Zhang, Baolong Bi, An Zhang, Ruipeng Wang, Tao Liang, and Xiang Wang. Neuron-level sequential editing for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16678–16702, 2025.

Omri Kaduri, Shai Bagon, and Tali Dekel. What's in the image? a deep-dive into the vision of vision language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 14549–14558, 2025.

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. See what you are told: Visual attention sink in large multimodal models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=7uDI7w5RQA.

Nora Kassner, Oyvind Tafjord, Hinrich Schütze, and Peter Clark. BeliefBank: Adding memory to a pre-trained language model for a systematic notion of belief. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 8849–8861, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.697. URL https://aclanthology.org/2021.emnlp-main.697/.

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

Kunhao Li, Wenhao Li, Di Wu, Lei Yang, Jun Bai, Ju Jia, and Jason Xue. Cross-modal unlearning via influential neuron path editing in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 35589–35597, 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, 2024.

Zherui Li, Houcheng Jiang, Hao Chen, Baolong Bi, Zhenhong Zhou, Fei Sun, Junfeng Fang, and Xiang Wang. Reinforced lifelong editing for language models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=1jUXprrfcb.

Zhihang Lin, Mingbao Lin, Luxi Lin, and Rongrong Ji. Boosting multimodal large language models with visual tokens withdrawal for rapid inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 5334–5342, 2025.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 2023.

Zheyuan Liu, Guangyao Dou, Xiangchi Yuan, Chunhui Zhang, Zhaoxuan Tan, and Meng Jiang. Modality-aware neuron pruning for unlearning in multimodal large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5913–5933, 2025.

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, et al. Deepseek-vl: towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*, 2024.

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, 2022.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=MkbcAHIYgyS.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/pdf?id=0DcZxeWfOPt.

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. PMLR, 2022b.

Shiwen Ni, Dingwei Chen, Chengming Li, Xiping Hu, Ruifeng Xu, and Min Yang. Forgetting before learning: Utilizing parametric arithmetic for knowledge updating in large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5716–5731, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.310. URL https://aclanthology.org/2024.acl-long.310/.

Haowen Pan, Yixin Cao, Xiaozhi Wang, Xun Yang, and Meng Wang. Finding and editing multimodal neurons in pre-trained transformers. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 1012–1037, 2024a.

Haowen Pan, Xiaozhi Wang, Yixin Cao, Zenglin Shi, Xun Yang, Juanzi Li, and Meng Wang. Precise localization of memories: A fine-grained neuron-level knowledge editing technique for LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=5xP1HDvpXI.

Kaihang Pan, Zhaoyu Fan, Juncheng Li, Qifan Yu, Hao Fei, Siliang Tang, Richang Hong, Hanwang Zhang, and Qianru Sun. Towards unified multimodal editing with enhanced knowledge collaboration. *Advances in Neural Information Processing Systems*, 37:110290–110314, 2024b.

Sarah Schwettmann, Neil Chowdhury, Samuel Klein, David Bau, and Antonio Torralba. Multimodal neurons in pretrained text-only transformers. In *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 2854–2859, 2023. doi: 10.1109/ICCVW60793.2023.00308.

Hoigi Seo, Dong Un Kang, Hyunjin Cho, Joohoon Lee, and Se Young Chun. On epistemic uncertainty of visual tokens for object hallucinations in large vision-language models. *Advances in Neural Information Processing Systems*, 38:130605–130660, 2026.

Jiajie Su, Haoyuan Wang, Xiaohua Feng, Yunshan Ma, Xiaobo Xia, Yuyuan Li, Xiaolin Zheng, Jianmao Xiao, and Chaochao Chen. Out-of-distribution generalization via invariant trajectories for multimodal large language model editing. *arXiv preprint arXiv:2601.19700*, 2026.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Jesse Vig and Yonatan Belinkov. Analyzing the structure of attention in a transformer language model. In Tal Linzen, Grzegorz Chrupała, Yonatan Belinkov, and Dieuwke Hupkes (eds.), *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 63–76, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4808. URL https://aclanthology.org/W19-4808/.

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. *Advances in Neural Information Processing Systems*, 37:53764–53797, 2024.

Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. Training networks in null space of feature covariance for continual learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pp. 184–193, 2021.

Zichuan Wang, Songlin Yang, Bo Peng, Zhenchen Tang, Yang Li, Beibei Dong, and Jing Dong. Same attention, different truths: Put logit-lens over visual attention to detect and mitigate lvlm object hallucination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25315–25325, 2026.

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. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 10222–10240, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.632. URL https://aclanthology.org/2023.emnlp-main.632/.

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*, 2024.

Zhi Zhang, Srishti Yadav, Fengze Han, and Ekaterina Shutova. Cross-modal information flow in multimodal large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 19781–19791, 2025.

Yang Zhou, Pengfei Cao, Yubo Chen, Qingbin Liu, Dianbo Sui, Xi Chen, Kang Liu, and Jun Zhao. M2Edit: Locate and edit multi-granularity knowledge in multimodal large language model. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 29029–29042, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1478. URL https://aclanthology.org/2025.emnlp-main.1478/.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. MiniGPT-4: Enhancing vision-language understanding with advanced large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=1tZbq88f27.

A Related Works

A.1 Vision Large Language Models

Vision Large Language Models (VLLMs) extend the capacity of large language models to multimodal perception by coupling a vision encoder, typically a Vision Transformer (ViT) (Dosovitskiy et al., 2021), with an LLM backbone through a lightweight modality projector. Early systems such as BLIP-2 (Li et al., 2023), LLaVA (Liu et al., 2023), and MiniGPT-4 (Zhu et al., 2024) demonstrate that aligning frozen vision features with powerful LLMs via limited instruction-tuning data is sufficient to elicit strong visual understanding, open-ended question answering, and instruction-following behavior. Subsequent efforts push this paradigm toward larger scale, denser vision–language alignment, and unified multimodal reasoning, as exemplified by recent proprietary and open families such as GLM (Hong et al., 2026), Qwen3-VL (Bai et al., 2025), and DeepSeek-VL (Lu et al., 2024). Despite their impressive performance, the factual knowledge encoded in these models inevitably drifts as the world changes, motivating knowledge-editing techniques that can surgically update specific beliefs without retraining the entire network. Our work targets this setting and, different from prior multimodal editors that depend on extra training, operates directly on the internal neurons of frozen VLLMs.

A.2 Knowledge Editing

Knowledge editing (KE) aims to update, override, or remove specific beliefs in a pretrained model while preserving its overall behavior, offering a lightweight alternative to full retraining. Existing approaches can be broadly grouped into three paradigms.

Memory-base and Meta-Learning Editors. A first line of work stores edits externally or learns to predict weight updates with a hypernetwork. Memory-based editors such as SERAC (Mitchell et al., 2022b) and belief-bank-style correction systems (Kassner et al., 2021; Huang et al., 2023; Dong et al., 2022) maintain a semi-parametric cache that overrides the model at inference time. Meta-learning editors instead train an auxiliary network to predict the parameter delta $\mathbf{\Delta}$: KE (De Cao et al., 2021) introduces a bidirectional LSTM editor under a latest-knowledge constraint, while MEND (Mitchell et al., 2022a) decomposes the editing gradient to make hypernetwork-based editing scalable to LLMs. To mitigate catastrophic forgetting, F-Learning (Ni et al., 2024) edits along a forgetting direction rather than fine-tuning toward new targets, and RLEdit (Li et al., 2025) casts lifelong editing as a reinforcement-learning problem for continuous updates. These methods, however, rely on a trained editor and therefore risk overfitting to specific data distributions.

Locate-Then-Edit. A second paradigm first identifies where a fact resides inside the model and then surgically modifies the corresponding parameters without any additional editor training. ROME (Meng et al., 2022) uses causal tracing to attribute factual recall to specific FFN modules and derives a closed-form rank-one update that maps a key representation to a new value. MEMIT (Meng et al., 2023) extends this procedure to batched multi-layer editing. PMET (Li et al., 2024) further incorporates multi-head self-attention into the localization to improve precision. Moving from layer-level to neuron-level localization, FiNE (Pan et al., 2025) computes per-neuron contribution scores and updates only the top-K neurons for each edit, sharply reducing computation while improving locality. To address sequential editing drift, NSE (Jiang et al., 2025) retains coherence by tracking "influential neurons" across edits. AlphaEdit (Fang et al., 2025) projects the weight update into the null space of preserved representations to minimize interference. Although these methods are interpretable and training-free, they are designed for text-only LLMs and degrade when visual tokens dominate the subject signal, as is the case in VLLMs.

Multimodal Knowledge Editing. Recent efforts extend KE to VLLMs but still rely on training additional parameters or external memories. WISE (Wang et al., 2024) learns a memory module with a unique abandonment-rectification mechanism for each edit; SMART (Pan et al., 2024b) patches self-distilled multimodal features through a trained editor; VisEdit (Chen et al., 2025b) trains an attribution-guided editor to adjust the patching location of the post-edit model; LiveEdit (Chen et al., 2025a) embeds a mixture-of-experts within the VLLM for streaming edits; and ODEdit (Su et al., 2026) treats multimodal editing as an out-of-distribution problem and uses a plug-and-play module.

While effective on in-distribution benchmarks, these editors tend to overfit their training distributions and degrade sharply on out-of-distribution edits, as also observed in our experiments. To the best of our knowledge, no prior *locate-then-edit* method edits multimodal knowledge by directly interpreting the internal signals of a VLLM, which is the gap **DELTA** fills.

A.3 Neuron Attribution

Understanding where factual knowledge resides inside a transformer has become a central theme in interpretability research. Dai et al. (2022) show that factual associations are stored as key–value memory in FFN neurons, and Geva et al. (2023) further dissect FFN blocks as key–value memories that are dynamically updated during inference. Subsequent work identifies knowledge-sensitive neurons via causal attribution (Pan et al., 2024a), providing the empirical foundation on which locate-then-edit methods build.

In the multimodal regime, this question becomes more nuanced because the relevant signal is distributed across visual and textual tokens. Schwettmann et al. (2023) and Liu et al. (2025) report that certain neurons respond consistently to the same concept regardless of whether it is conveyed through an image or a word, suggesting the existence of cross-modal concept neurons. Building on this observation, NAM (Fang et al., 2024) proposes the first neuron attribution framework specifically designed for multimodal LLMs: it scores each FFN neuron k at layer ℓ for an output token t by $s^{(\ell)}(k,t) = a_k^{(\ell)} \langle \boldsymbol{w}_{\text{down}}^{(\ell),k}, \boldsymbol{r}_t \rangle$, where $a_k^{(\ell)}$ is the neuron activation and $\boldsymbol{r}_t$ is the target representation, and decouples the score into image- and text-only components to expose *image-neurons* (*I-neurons*), *text-neurons* (*T-neurons*), and cross-modal neurons. NAM is gradient-free and an order of magnitude faster than attribution methods based on integrated gradients, while still supporting downstream multimodal knowledge editing.

A complementary direction emphasizes inter-layer structure rather than per-neuron scores. MIP-Editor (Li et al., 2026) argues that point-wise activation scoring cannot capture the structured information flow across layers, and proposes an *influential neuron path* that traces a coherent chain of neurons from low to high layers. The path is selected by accumulating layer-wise contributions, and editing is realized by representation misdirection that pushes the forgotten-data representation away from its original semantic direction.

Our work builds on these insights but differs in two respects. First, instead of relying on a single point-wise score or a pre-defined cross-layer path, **DELTA** couples modality-specific activations with a layer-wise fusion condition, capturing both intra-layer modality decoupling and cross-layer coherence. Second, while NAM attributes neurons to an output token and MIP-Editor attributes them along an unlearning path, **DELTA** attributes neurons to the edit target itself, producing a localized editing set that is well-suited for the subsequent null-space constrained update.

B Hidden States Computation and Neurons Decomposition in VLLMs

We provide a formal derivation of the VLLM's hidden-state computation and per-neuron FFN decomposition, which grounds the trace-based editing described in the main text.

VLLM Architecture. Let $f_{\mathcal{W}}$ be the VLLM, composed of a vision encoder, a modality projector, and an LLM backbone with L transformer layers, hidden dimension d_h, and FFN intermediate dimension d_{ffn}. Given an image–text pair $(\boldsymbol{I}, \boldsymbol{p})$, the vision encoder and projector map the image into N_v vision-token embeddings $\boldsymbol{z}_{\text{vis}} \in \mathbb{R}^{N_v \times d_h}$, while the text prompt is tokenized and embedded into N_t text-token embeddings $\boldsymbol{z}_{\text{txt}} \in \mathbb{R}^{N_t \times d_h}$. They are concatenated along the sequence dimension to form the initial hidden state

$$\boldsymbol{h}^{(0)} = [\boldsymbol{z}_{\text{vis}};\, \boldsymbol{z}_{\text{txt}}] \in \mathbb{R}^{S \times d_h}, \qquad S = N_v + N_t. \tag{15}$$

For autoregressive generation of a target sequence $\boldsymbol{y}_e = (\hat{y}_1, \ldots, \hat{y}_T)$, the already-generated prefix is appended analogously: $\boldsymbol{h}_t^{(0)} = [\boldsymbol{z}_{\text{vis}};\, \boldsymbol{z}_{\text{txt}};\, \boldsymbol{z}_{\hat{y}_{<t}}] \in \mathbb{R}^{(S+t-1) \times d_h}$, where $\boldsymbol{z}_{\hat{y}_{<t}}$ stacks the embeddings of $\hat{y}_1, \ldots, \hat{y}_{t-1}$.

Layer-wise Recurrence. For each layer $\ell \in \{1, \dots, L\}$, the hidden state evolves via the standard residual update (Eqn. 1):

$$\boldsymbol{h}^{(\ell)} = \boldsymbol{h}^{(\ell-1)} + \text{Attn}^{(\ell)}\big(\boldsymbol{h}^{(\ell-1)}\big) + \text{FFN}^{(\ell)}\big(\tilde{\boldsymbol{h}}^{(\ell-1)}\big), \tag{16}$$

where $\tilde{\boldsymbol{h}}^{(\ell-1)} = \boldsymbol{h}^{(\ell-1)} + \text{Attn}^{(\ell)}(\boldsymbol{h}^{(\ell-1)})$ is the post-attention intermediate state, and the FFN at layer ℓ is the two-layer MLP

$$\begin{aligned} \text{FFN}^{(\ell)}\big(\tilde{\boldsymbol{h}}^{(\ell-1)}\big) &= \mathbf{a}^{(\ell)}\, \mathbf{W}_{\text{down}}^{(\ell)\,\top} \in \mathbb{R}^{S \times d_h}, \\ \mathbf{a}^{(\ell)} &= \sigma_{\text{ffn}}\big(\gamma(\tilde{\boldsymbol{h}}^{(\ell-1)})\, \mathbf{W}_{\text{up}}^{(\ell)}\big) \in \mathbb{R}^{S \times d_{\text{ffn}}}, \end{aligned} \tag{17}$$

with $\mathbf{W}_{\text{up}}^{(\ell)}, \mathbf{W}_{\text{down}}^{(\ell)} \in \mathbb{R}^{d_h \times d_{\text{ffn}}}$, internal layer norm $\gamma(\cdot)$ applied in pre-norm style before the up-projection (e.g., `final_layer_norm` in BLIP2-OPT), and σ_{ffn} the FFN non-linearity (ReLU for OPT-based backbones, GELU for LLaMA-based backbones).

Token-level View. For any position $i \in [1, S]$, the per-token hidden state $\boldsymbol{h}_i^{(\ell)} = [\boldsymbol{h}^{(\ell)}]_{i,:} \in \mathbb{R}^{d_h}$ and its FFN output decompose as

$$\boldsymbol{h}_i^{(\ell)} = \boldsymbol{h}_i^{(\ell-1)} + \text{Attn}_i^{(\ell)}(\boldsymbol{h}^{(\ell-1)}) + \sum_{j=1}^{d_{\text{ffn}}} a_{i,j}^{(\ell)}\, \boldsymbol{w}_{\text{down}}^{(\ell),j}, \tag{18}$$

where $a_{i,j}^{(\ell)} = [\mathbf{a}^{(\ell)}]_{i,j} = \sigma_{\text{ffn}}\big(\langle [\gamma(\tilde{\boldsymbol{h}}^{(\ell-1)})]_{i,:},\, \boldsymbol{w}_{\text{up}}^{(\ell),j} \rangle\big)$ is the post-activation score of neuron (ℓ, j) at position i, and $\boldsymbol{w}_{\text{up}}^{(\ell),j}, \boldsymbol{w}_{\text{down}}^{(\ell),j} \in \mathbb{R}^{d_h}$ are the j-th columns of $\mathbf{W}_{\text{up}}^{(\ell)}, \mathbf{W}_{\text{down}}^{(\ell)}$ respectively. We partition $[1, S]$ into $\mathcal{P}_{\text{vis}} = [1, N_v]$, $\mathcal{P}_{\text{txt}} = [N_v{+}1, S]$, and the subject range $\mathcal{P}_{\text{targ}} \subseteq \mathcal{P}_{\text{txt}}$ (Eqn. 3).

Unrolling the Recurrence. By telescoping the residual recurrence, the hidden state at the final layer can be written as the cumulative sum of every component contribution:

$$\boldsymbol{h}^{(L)} = \boldsymbol{h}^{(0)} + \sum_{\ell=1}^{L} \text{Attn}^{(\ell)}\big(\boldsymbol{h}^{(\ell-1)}\big) + \sum_{\ell=1}^{L} \text{FFN}^{(\ell)}\big(\tilde{\boldsymbol{h}}^{(\ell-1)}\big). \tag{19}$$

The hidden state at the last prompt position, $\boldsymbol{h}_{S,:}^{(L)} \in \mathbb{R}^{d_h}$, is projected to vocabulary logits via the unembedding matrix $\mathbf{W}_u \in \mathbb{R}^{V \times d_h}$. For any candidate token $y \in [1, V]$, denote its unembedding row by $\mathbf{w}_u[y] = \mathbf{W}_u[y, :] \in \mathbb{R}^{d_h}$; then

$$\text{logit}_S(y) = \mathbf{w}_u[y]^\top \boldsymbol{h}_{S,:}^{(L)}, \qquad P(y \mid \boldsymbol{I}, \boldsymbol{p}) = \text{softmax}\big(\text{logit}_S\big)_y. \tag{20}$$

For a generated target token $\hat{y}_t$ at the t-th decoding step, the same readout applies at position $S{+}t{-}1$ of the extended sequence $\boldsymbol{h}_t^{(0)}$.

Per-Neuron Activation and Contribution. Stacking the per-token decomposition across all S positions, the FFN output at layer ℓ admits the sequence-level rank-d_{ffn} form

$$\text{FFN}^{(\ell)}\big(\tilde{\boldsymbol{h}}^{(\ell-1)}\big) = \sum_{j=1}^{d_{\text{ffn}}} \mathbf{a}_{:,j}^{(\ell)}\, \boldsymbol{w}_{\text{down}}^{(\ell),j\,\top} \in \mathbb{R}^{S \times d_h}, \tag{21}$$

where $\mathbf{a}_{:,j}^{(\ell)} \in \mathbb{R}^S$ is the j-th column of $\mathbf{a}^{(\ell)}$—this is precisely the post-activation FFN intermediate whose capture via a forward pre-hook on $\mathbf{W}_{\text{down}}^{(\ell)}$ is described in Section 3.1 (Eqn. 1). Substituting into the unrolled recurrence and reading out the last position $i = S$ gives

$$\boldsymbol{h}_{S,:}^{(L)} = \boldsymbol{h}_{S,:}^{(0)} + \sum_{\ell=1}^{L} \text{Attn}_S^{(\ell)}(\boldsymbol{h}^{(\ell-1)}) + \sum_{\ell=1}^{L} \sum_{j=1}^{d_{\text{ffn}}} a_{S,j}^{(\ell)}\, \boldsymbol{w}_{\text{down}}^{(\ell),j}, \tag{22}$$

where $\text{Attn}_S^{(\ell)}(\cdot)$ denotes the attention output at the last prompt position. The same expansion holds at any position $i \in [1, S]$.

Trace-Neuron Masking and Edit Impact. The localized target neurons $\mathcal{T} = \cup_{\ell \in \mathcal{L}} \mathcal{T}_\ell \subset \mathcal{N}$ induce a per-layer binary mask over FFN neurons. We decompose the FFN output at each layer into trace and non-trace contributions:

$$\sum_{j=1}^{d_{\text{ffn}}} a_{i,j}^{(\ell)} \, \boldsymbol{w}_{\text{down}}^{(\ell),j} = \underbrace{\sum_{j \in \mathcal{T}_\ell} a_{i,j}^{(\ell)} \, \boldsymbol{w}_{\text{down}}^{(\ell),j}}_{\text{trace neurons}} + \underbrace{\sum_{j \notin \mathcal{T}_\ell} a_{i,j}^{(\ell)} \, \boldsymbol{w}_{\text{down}}^{(\ell),j}}_{\text{non-trace neurons}}, \tag{23}$$

where $\mathcal{T}_\ell = \{j \mid (\ell, j) \in \mathcal{T}\}$. Editing only modifies the trace neurons' down-projection columns via $\boldsymbol{w}_{\text{down}}^{(\ell),j} \leftarrow \boldsymbol{w}_{\text{down}}^{(\ell),j} + \boldsymbol{\delta}_{\ell,j}$ for $j \in \mathcal{T}_\ell$ (Eqn. 10). The induced logit change at position i for any candidate token $y \in [1, V]$ is

$$\Delta\text{logit}_i(y) \;=\; \sum_{\ell \in \mathcal{L}} \sum_{j \in \mathcal{T}_\ell} a_{i,j}^{(\ell)} \, \mathbf{w}_u[y]^\top \boldsymbol{\delta}_{\ell,j}, \tag{24}$$

which is confined to the trace neurons' activation-weighted delta projections, providing the formal basis for exclusive back-propagation to $\mathcal{T}$ in Section 4. For the editing position $i = S$ (or $i = S{+}t{-}1$ during generation), Eqn. 24 drives the model toward $\boldsymbol{y}_e$; for $i \notin \mathcal{P}_{\text{targ}}$, the null-space constraint of Section 4.3 further bounds $\Delta\text{logit}_i(y)$ to preserve locality (detailed in Appendix C).

C Detailed Derivation of Null-Space Regularized Optimization

We provide a self-contained derivation of the null-space constrained optimisation that underpins DELTA's locality preservation. Unlike AlphaEdit (Fang et al., 2025), which constructs the null space from an external preservation dataset, DELTA derives the constraint solely from the edit input $(\boldsymbol{I}_e, \boldsymbol{x}_e)$ itself. The key insight is that the activation pattern of the edit input already spans the directions that must be preserved for locality; directions *not* excited by the edit input are free to be modified. We formalise this below in four steps that together form a closed theoretical chain from the original locality requirement to the final weight update.

FFN output under delta injection. Recall from Section 4 that for layer $\ell \in \mathcal{L}$ with trace-neuron set $\mathcal{T}_\ell$, the per-neuron deltas are stacked column-wise into $\boldsymbol{\Delta}^{(\ell)} = [\boldsymbol{\delta}_{\ell,j}]_{j \in \mathcal{T}_\ell} \in \mathbb{R}^{d_h \times |\mathcal{T}_\ell|}$, and they modify the FFN output as (Eqn. 10)

$$\text{FFN}_{\boldsymbol{\delta}}^{(\ell)}(\tilde{\boldsymbol{h}}^{(\ell-1)}) = \mathbf{a}^{(\ell)} {\mathbf{W}_{\text{down}}^{(\ell)}}^\top + \mathbf{a}_{:,\mathcal{T}_\ell}^{(\ell)} \, \boldsymbol{\Delta}^{(\ell)^\top}. \tag{25}$$

Let $\mathbf{A}^{(\ell)} = \mathbf{a}_{:,\mathcal{T}_\ell}^{(\ell)} \in \mathbb{R}^{S \times |\mathcal{T}_\ell|}$ be the activation sub-matrix at trace neurons under the edit input. The perturbation to the layer-ℓ FFN output, evaluated on the edit input, is

$$\delta\text{FFN}^{(\ell)} = \mathbf{A}^{(\ell)} \boldsymbol{\Delta}^{(\ell)^\top} \in \mathbb{R}^{S \times d_h}. \tag{26}$$

Preservation criterion. For locality, we require the edited model to reproduce the original FFN output on the edit input up to linear order, i.e. $\delta\text{FFN}^{(\ell)} \approx \mathbf{0}$. By equation 26, this is equivalent to the matrix-level condition

$$\mathbf{A}^{(\ell)} \boldsymbol{\Delta}^{(\ell)^\top} \approx \mathbf{0}_{S \times d_h}, \tag{27}$$

i.e. the (right) null space of $\mathbf{A}^{(\ell)} \in \mathbb{R}^{S \times |\mathcal{T}_\ell|}$ must contain every column of $\boldsymbol{\Delta}^{(\ell)^\top}$. Each such column is a $|\mathcal{T}_\ell|$-vector that collects, across the trace neurons $j \in \mathcal{T}_\ell$, one fixed output coordinate of the per-neuron deltas $\boldsymbol{\delta}_{\ell,j} \in \mathbb{R}^{d_h}$ stacked column-wise in $\boldsymbol{\Delta}^{(\ell)}$ (Eqn. 10); there are d_h such columns, one per coordinate of the FFN output space. Since $\boldsymbol{\Delta}^{(\ell)}$ must simultaneously satisfy the editing objective (Section 4), a closed-form solution is unavailable; we instead incorporate this requirement into the optimisation via the derivation below.

Step 1 — Optimisation formulation. We formalise equation 27 as minimising the total activation perturbation in the Frobenius norm:

$$\min_{\{\boldsymbol{\Delta}^{(\ell)}\}_{\ell \in \mathcal{L}}} \sum_{\ell \in \mathcal{L}} \left\| \mathbf{A}^{(\ell)} \boldsymbol{\Delta}^{(\ell)^\top} \right\|_F^2. \tag{28}$$

This is the natural squared-loss encoding of equation 27: the squared Frobenius norm decomposes column-wise over $\boldsymbol{\Delta}^{(\ell)^\top}$, summing $\|\mathbf{A}^{(\ell)}\boldsymbol{c}_k\|_2^2$ across the d_h columns $\boldsymbol{c}_k \in \mathbb{R}^{|\mathcal{T}_\ell|}$ of $\boldsymbol{\Delta}^{(\ell)^\top}$. The objective is everywhere differentiable, permitting gradient-based optimisation jointly with the other loss terms in $\mathcal{L}_{\boldsymbol{\Delta}}$ (Section 4.2).

Step 2 — Reduction to activation covariance. Expanding the Frobenius norm:

$$\left\|\mathbf{A}^{(\ell)}\boldsymbol{\Delta}^{(\ell)^\top}\right\|_F^2 = \mathrm{tr}\Big(\boldsymbol{\Delta}^{(\ell)}\,\mathbf{A}^{(\ell)^\top}\mathbf{A}^{(\ell)}\,\boldsymbol{\Delta}^{(\ell)^\top}\Big). \tag{29}$$

Thus the problem is fully determined by the activation covariance matrix

$$\mathbf{C}^{(\ell)} = \mathbf{A}^{(\ell)^\top}\mathbf{A}^{(\ell)} \in \mathbb{R}^{|\mathcal{T}_\ell|\times|\mathcal{T}_\ell|}. \tag{30}$$

The (j, j')-th entry $\sum_{i=1}^{S} A_{i,j}^{(\ell)} A_{i,j'}^{(\ell)}$ measures the co-activation of trace neurons j and j' across the edit-input token positions; larger entries indicate neuron pairs that are jointly excited by the edit input, and directions with near-zero covariance correspond to neuron combinations that the edit input barely activates. This reduction from $\mathbf{A}^{(\ell)}$ to $\mathbf{C}^{(\ell)}$ is the key insight: the quadratic form absorbs the S sequence positions, leaving a $|\mathcal{T}_\ell| \times |\mathcal{T}_\ell|$ matrix whose spectral structure directly encodes which linear combinations of trace neurons are excited by the edit input and which are not. Note that $\mathbf{C}^{(\ell)}$ uses *unweighted* token positions; protection of the visual manifold is enforced separately and orthogonally by the stochastic-perturbation loss $\mathcal{L}_{\text{vis}}$ of Section 4.2, which prevents $\boldsymbol{\Delta}^{(\ell)}$ from overfitting to the exact realisation of $\boldsymbol{I}_e$ rather than entering the covariance itself.

Step 3 — Spectral decomposition and null-space projector. Let $\mathbf{A}^{(\ell)} = \mathbf{U}^{(\ell)}\boldsymbol{\Sigma}^{(\ell)}\mathbf{V}^{(\ell)^\top}$ be the SVD, with singular values ordered ascending $\sigma_1^{(\ell)} \le \cdots \le \sigma_{|\mathcal{T}_\ell|}^{(\ell)}$, so that

$$\mathbf{C}^{(\ell)} = \mathbf{A}^{(\ell)^\top}\mathbf{A}^{(\ell)} = \mathbf{V}^{(\ell)}\big(\boldsymbol{\Sigma}^{(\ell)}\big)^2\mathbf{V}^{(\ell)^\top}, \tag{31}$$

so the right singular vectors of $\mathbf{A}^{(\ell)}$ are exactly the eigenvectors of $\mathbf{C}^{(\ell)}$, with eigenvalues $(\sigma_i^{(\ell)})^2$. This equivalence holds because $\|\mathbf{A}^{(\ell)}\boldsymbol{\delta}\|_2^2 = \boldsymbol{\delta}^\top\mathbf{C}^{(\ell)}\boldsymbol{\delta}$: the quadratic form is identical whether one works with $\mathbf{A}^{(\ell)}$ or $\mathbf{C}^{(\ell)}$, so minimising the activation perturbation is equivalent to finding directions that minimise the Rayleigh quotient of $\mathbf{C}^{(\ell)}$. Consequently, solving for the near-null space of $\mathbf{A}^{(\ell)}$ reduces to finding the eigenvectors of $\mathbf{C}^{(\ell)}$ with the smallest eigenvalues—the computation involves only the $|\mathcal{T}_\ell| \times |\mathcal{T}_\ell|$ covariance rather than the $S \times |\mathcal{T}_\ell|$ activation matrix.

We collect the d_{null} right singular vectors corresponding to the smallest d_{null} singular values into

$$\mathbf{V}_\perp^{(\ell)} = [\mathbf{v}_1^{(\ell)}, \ldots, \mathbf{v}_{d_{\text{null}}}^{(\ell)}] \in \mathbb{R}^{|\mathcal{T}_\ell|\times d_{\text{null}}}. \tag{32}$$

The orthogonal projector onto this near-null space is

$$\mathbf{P}_{d_{\text{null}}}^{(\ell)} = \mathbf{V}_\perp^{(\ell)}\mathbf{V}_\perp^{(\ell)^\top} \in \mathbb{R}^{|\mathcal{T}_\ell|\times|\mathcal{T}_\ell|}, \tag{33}$$

which is symmetric and idempotent ($(\mathbf{P}_{d_{\text{null}}}^{(\ell)})^2 = \mathbf{P}_{d_{\text{null}}}^{(\ell)}$). Its complement $\mathbf{I} - \mathbf{P}_{d_{\text{null}}}^{(\ell)}$ projects onto the rowspace of $\mathbf{C}^{(\ell)}$—the directions the edit input actively excites and that must be preserved.

Step 4 — Soft regularisation and hard projection. The rowspace component of $\boldsymbol{\Delta}^{(\ell)^\top}$ is $(\mathbf{I} - \mathbf{P}_{d_{\text{null}}}^{(\ell)})\,\boldsymbol{\Delta}^{(\ell)^\top}$. During optimisation we penalise its squared Frobenius norm (Eqn. 13):

$$\boxed{\mathcal{L}_{\text{ns}} = \sum_{\ell\in\mathcal{L}} \left\|\big(\mathbf{I} - \mathbf{P}_{d_{\text{null}}}^{(\ell)}\big)\,\boldsymbol{\Delta}^{(\ell)^\top}\right\|_F^2}. \tag{34}$$

Since $\mathbf{I} - \mathbf{P}_{d_{\text{null}}}^{(\ell)}$ is an orthogonal projector, the squared norm expands as $\mathrm{tr}\big(\boldsymbol{\Delta}^{(\ell)}\,(\mathbf{I} - \mathbf{P}_{d_{\text{null}}}^{(\ell)})\,\boldsymbol{\Delta}^{(\ell)^\top}\big)$, yielding the gradient $\nabla_{\boldsymbol{\Delta}^{(\ell)^\top}}\mathcal{L}_{\text{ns}} = 2(\mathbf{I} - \mathbf{P}_{d_{\text{null}}}^{(\ell)})\,\boldsymbol{\Delta}^{(\ell)^\top}$, equivalently $\nabla_{\boldsymbol{\Delta}^{(\ell)}}\mathcal{L}_{\text{ns}} = 2\boldsymbol{\Delta}^{(\ell)}(\mathbf{I} - \mathbf{P}_{d_{\text{null}}}^{(\ell)})$ when differentiated in the same orientation as $\boldsymbol{\Delta}^{(\ell)}$. The squared form is preferred over the unsquared variant because it avoids the numerical instability of $\boldsymbol{x}/\|\boldsymbol{x}\|$ near zero and guarantees a

well-conditioned gradient everywhere. This soft constraint guides the optimiser toward null-space-compatible solutions without requiring exact membership during training.

After $\mathcal{L}_{\boldsymbol{\Delta}}$ converges, the remaining rowspace component is discarded via a one-time hard projection:

$$\boxed{\boldsymbol{\Delta}^{(\ell)} \leftarrow \boldsymbol{\Delta}^{(\ell)}\,\mathbf{P}^{(\ell)}_{d_{\text{null}}}}, \tag{35}$$

which right-multiplies each row of $\boldsymbol{\Delta}^{(\ell)}$ by $\mathbf{P}^{(\ell)}_{d_{\text{null}}}$, removing all rowspace components and retaining only the near-null-space component. Applying the projector yields

$$\mathbf{A}^{(\ell)}\big(\boldsymbol{\Delta}^{(\ell)}\mathbf{P}^{(\ell)}_{d_{\text{null}}}\big)^{\top} = \mathbf{A}^{(\ell)}\mathbf{P}^{(\ell)}_{d_{\text{null}}}\boldsymbol{\Delta}^{(\ell)\top} \approx \mathbf{0}, \tag{36}$$

so the FFN output on the edit input is preserved at the linearised level. The projected deltas are then written permanently into the down-projection weights:

$$\boldsymbol{w}^{(\ell),j}_{\text{down}} \leftarrow \boldsymbol{w}^{(\ell),j}_{\text{down}} + \boldsymbol{\delta}_{\ell,j}, \qquad (\ell, j) \in \mathcal{T}. \tag{37}$$

Non-trace weights ($j \notin \mathcal{T}_\ell$) are untouched. The forward hook of Eqn. 10 is removed before the write step, so subsequent forward passes—including the next edit in a lifelong sequence—see the updated $\mathbf{W}^{(\ell)}_{\text{down}}$, closing the editing loop.

Locality guarantee. For any vector $\boldsymbol{v} \in \mathbb{R}^{|\mathcal{T}_\ell|}$ in $\text{span}(\mathbf{V}^{(\ell)}_{\perp})$—i.e. any $|\mathcal{T}_\ell|$-dimensional direction across trace neurons, such as one of the columns of $\boldsymbol{\Delta}^{(\ell)^{\top}}$ after the hard projection equation 35—expand it in the right singular basis: $\boldsymbol{v} = \sum_{i=1}^{d_{\text{null}}} c_i \mathbf{v}^{(\ell)}_i$. Then

$$\big\|\mathbf{A}^{(\ell)}\boldsymbol{v}\big\|_2^2 = \boldsymbol{v}^{\top}\mathbf{C}^{(\ell)}\boldsymbol{v} = \sum_{i=1}^{d_{\text{null}}} (\sigma^{(\ell)}_i)^2\, c_i^2 \le (\sigma^{(\ell)}_{d_{\text{null}}})^2\, \|\boldsymbol{v}\|_2^2, \tag{38}$$

where $\sigma^{(\ell)}_{d_{\text{null}}}$ is the largest among the d_{null} retained (i.e. smallest) singular values. Consequently $\mathbf{A}^{(\ell)}\boldsymbol{v} \approx \mathbf{0}$, with the approximation error bounded by $\sigma^{(\ell)}_{d_{\text{null}}}\|\boldsymbol{v}\|_2$, which is small by construction since the retained directions correspond to the smallest singular values. The dual-stage design—soft regularisation equation 34 followed by hard projection equation 35—provides both training stability (the soft penalty prevents the optimiser from drifting into rowspace directions that must later be projected away) and an approximate post-hoc locality guarantee (the final projection bounds the activation perturbation by the truncation error of the spectral decomposition).

Relation to AlphaEdit. AlphaEdit (Fang et al., 2025) constructs an analogous null space from an auxiliary preservation dataset $\mathcal{D}_{\text{preserve}}$, requiring an additional forward pass over irrelevant inputs. DELTA replaces $\mathcal{D}_{\text{preserve}}$ with the edit input itself: since the edit input already encodes the visual and textual patterns that the edited model must continue to handle correctly, directions it does *not* activate are precisely those that can be modified without side effects. This eliminates the need for extra data while providing the same linearised locality guarantee; robustness to visual variation (perspective, color temperature, etc.) is handled orthogonally by the stochastic-perturbation loss $\mathcal{L}_{\text{vis}}$ (Section 4.2), which regularises $\boldsymbol{\Delta}^{(\ell)}$ against perturbed vision-token embeddings $\widetilde{\boldsymbol{z}}_{\text{vis}} = \boldsymbol{z}_{\text{vis}} + \boldsymbol{\xi}$ and prevents the null-space solution from overfitting to the exact realisation of $\boldsymbol{I}_e$.

D Implementation Details: FFN Block, Hook Mechanics, and Module Paths

We list all parameter settings below. We report the candidate layers corresponding to each VLLM backbone used in our experiments, visualization, and ablation studies. Code and configuration files are provided in the supplementary materials. The candidate layers are derived empirically from the (C1)–(C3) conditions, as detailed in Appendix G. The following settings produces the results of Tab. 1, Tab. 2, Tab. 3, etc. in the paper main body.

BLIP2-OPT-2.7B BLIP2-OPT-2.7B backbone is localized with candidate layers $\mathcal{L} = \{17, 18, 19, 26, 27, 28, 29, 30, 31\}$ (9 layers) with top $K = 200$ neurons per layer selected, where $K_{\text{vis}} = K_{\text{targ}} = 20$ and the optimization epochs $E = 20$, $d_{\text{null}} = 100$.

LLaVA-1.5-7B LLaVA-1.5-7B backbone is lcoalized with candidate layers $\mathcal{L} = \{1, 19, 20, 21, 22, 23, 24, 25, 26, 27, 30\}$ (11 layers) with top $K = 800$ neurons per layer selected, where $K_{\text{vis}} = K_{\text{targ}} = 20$ and the optimization epochs $E = 25$, $d_{\text{null}} = 100$.

MiniGPT-4-Vicuna-7B MiniGPT-4-Vicuna-7B backbone is localized with candidate layers $\mathcal{L} = \{21, 22, 23, 24, 26, 27, 28, 29\}$ (8 layers) with top $K = 1500$ neurons per layer seleted, where $K_{\text{vis}} = K_{\text{targ}} = 20$ and the optimization epochs $E = 20$, $d_{\text{null}} = 100$.

E More Experimental Results on Other VLLM Backbones

In this section, we provide additional experiments on other VLLM backbones (Qwen2.5-VL-3B-Instruct and Qwen3-VL-8B-Instruct (Bai et al., 2025)). As shown in Tab. 4, for a more intuisive display, we directly compare the multimodal knowledge editing results under the same locate-then-edit paradigm. We find that our **DELTA** also outperforms other approaches on the latest VLLM backbones with significant advancement. Though FiNE seems to achieve a higher T-Gen/M-Gen scores, it always performs with much lower T-Loc/M-Loc, which means the significant changes on the area unrelated to the current knowledge.

Method	EIC						EVQA						VLKEB					
	Rel.	T-Gen.	M-Gen.	T-Loc.	M-Loc.	*Avg.*	Rel.	T-Gen.	M-Gen.	T-Loc.	M-Loc.	*Avg.*	Rel.	T-Gen.	M-Gen.	T-Loc.	M-Loc.	*Avg.*
Qwen2.5-VL (3B)																		
MEMIT	0.45	0.16	0.07	96.19	90.17	37.41 (±0.00)	1.47	2.00	1.60	83.33	51.83	28.05 (±0.00)	0.00	0.00	0.00	96.67	100.00	39.33 (±0.00)
PMET	0.40	0.18	0.04	98.59	96.42	39.13 (±0.00)	0.00	0.00	0.00	97.91	90.83	37.75 (±0.00)	0.00	0.00	0.00	100.00	100.00	40.00 (±0.00)
AlphaEdit	0.34	0.16	0.05	98.96	96.94	39.29 (±0.00)	0.03	0.13	0.20	0.9665	0.9687	45.90 (±0.00)	0.00	0.00	0.00	98.89	95.00	38.78 (±0.00)
FiNE	99.97	93.20	96.27	31.01	5.31	65.15 (±0.00)	100.00	95.33	72.78	58.95	20.90	69.59 (±0.00)	100.00	94.36	98.62	46.41	20.79	72.04 (±0.00)
DELTA	99.92	78.89	79.88	81.13	80.24	84.01 (±0.18)	100.00	94.70	70.90	77.90	74.36	83.57 (±0.24)	98.85	77.68	77.03	91.63	81.14	85.27 (±0.26)
Qwen3-VL (8B)																		
FiNE	100.00	98.02	97.18	18.18	2.87	63.26 (±0.23)	100.00	98.03	88.98	64.67	23.74	75.08 (±0.16)	100.00	96.67	97.95	43.72	25.07	72.68 (±0.25)
DELTA	100.00	81.90	72.45	81.80	75.00	**82.23** (±0.31)	100.00	95.20	74.21	83.10	81.06	**86.31** (±0.22)	98.89	76.08	77.53	91.18	81.69	**85.07** (±0.27)

Table 4: We perform single-edit across 3 datasets. We evaluate typical locate-then-edit methods on Qwen2.5-VL-3B-Instruct and a larger scale VLLM (Qwen3-VL-8B-Instruct).

F More Visualization Results of Instance Analysis

In this section, we provide additional visualization results illustrating how DELTA attributes neurons to visual regions, as shown in Fig. 8, Fig. 9, Fig. 10, and Fig. 11. For each editing target neuron, we record which tokens contribute to it. The more neurons a visual region contributes to, the higher its score. We visualize these areas with highlight colors in the style of attention msks (shown in the following several pages).

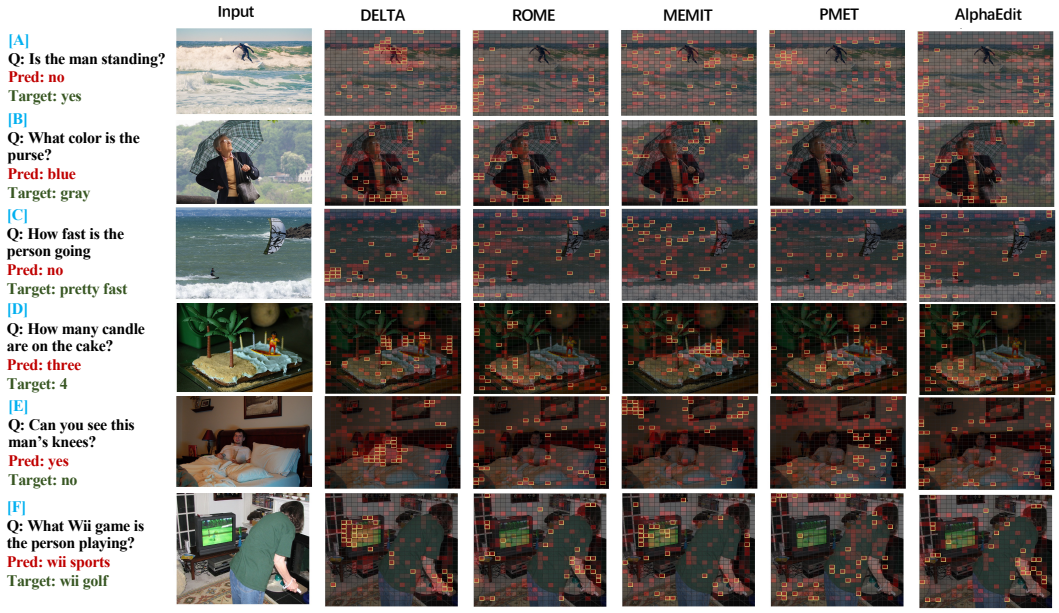

Figure 8: Instance visualization of the target neurons. In case (A)–(F), only DELTA focus the key area of the input image.

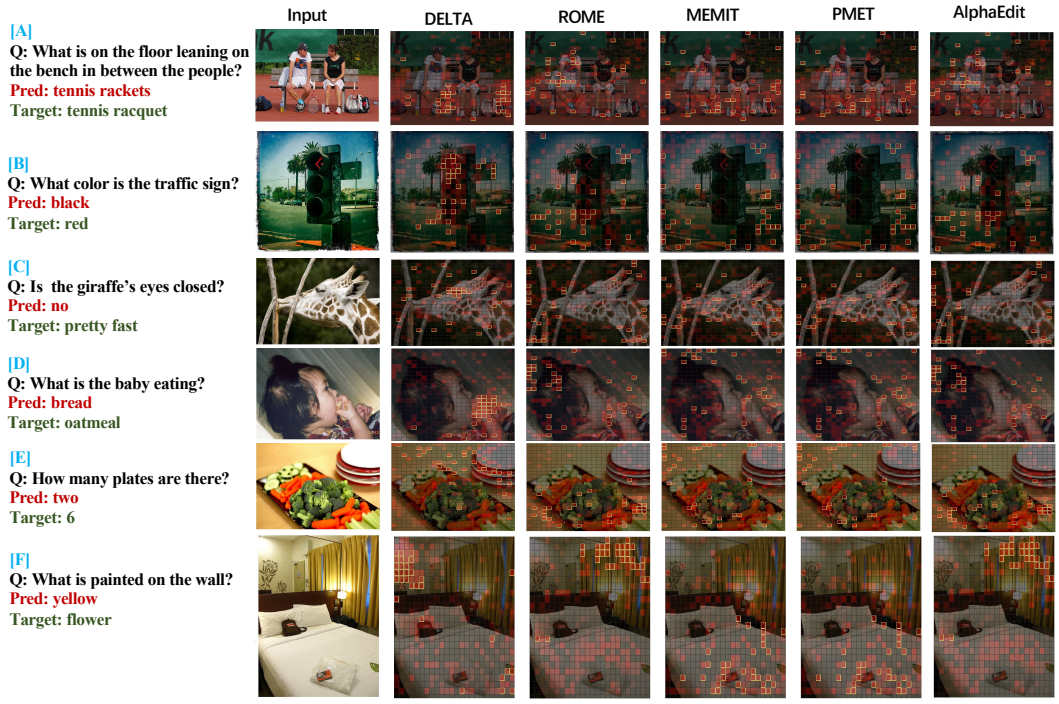

Figure 9: Instance visualization of the target neurons. In case (A)–(F), only DELTA focus the key area of the input image.

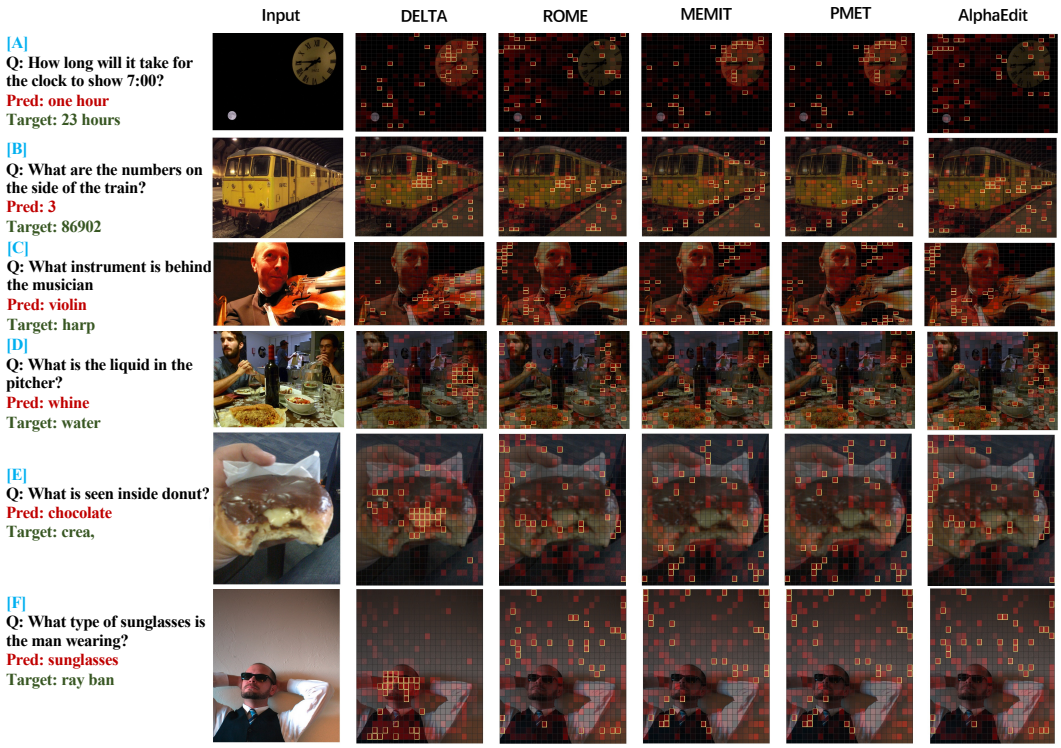

Figure 10: Instance visualization of the target neurons. In case (A)–(F), only DELTA focus the key area of the input image.

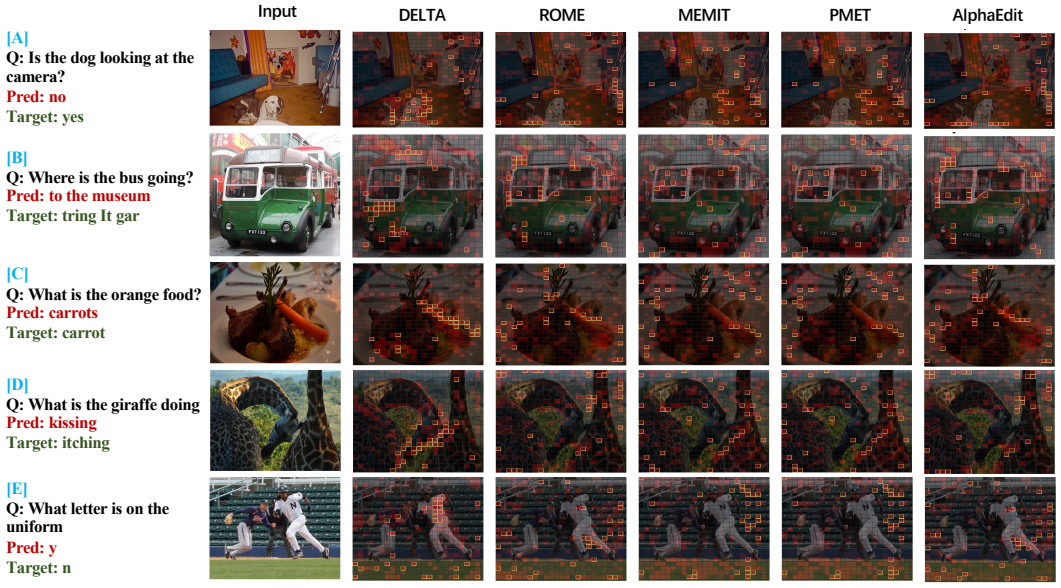

Figure 11: Instance visualization of the target neurons. In case (A)–(E), only DELTA focus the key area of the input image.

G More Localization Samples Results and Analysis

In this section, we revisit the question raised in "Localization Discussions" in conjunction with Fig. 2. We sequentially visualize the localization results of DELTA under the (C1)–(C3) conditions. The results are presented in Fig. 12, Fig. 13, Fig. 14, Fig. 15, Fig. 16, Fig. 17, Fig. 18, Fig. 19, Fig. 20, and Fig. 21. Note that this supplementary file is compressed, for high queliy localization curves, please refer to the "Media Supplement". We upload the original PDF there together with instance analysis results.

As hypothesized in the main paper, multimodal (predominantly visual) knowledge is first abstracted and then repeatedly re-integrated into the textual residual stream at a sparse subset of layers. Applying our three-condition localization criterion (C1)–(C3) reveals a pronounced asymmetry in this signal: the resulting layer sets differ substantially across architectures, yet remain remarkably stable across different edit samples within the same backbone. In other words, when the three diagnostic indicators are tracked throughout the forward pass on a fixed model with completely different edit instances, they exhibit nearly identical layer-wise trends.

BLIP2-OPT. In BLIP2-OPT-2.7B (Fig. 12 and Fig. 13), I_{mean} and I_{entropy} increase monotonically with depth, whereas κ_ℓ exhibits an initial surge followed by a rapid decay and a gentle recovery over the final layers. This behavior is consistent with the BLIP2 architecture, in which a Q-Former abstracts visual information and injects it into the textual hidden states. Correspondingly, BLIP2-OPT-2.7B is consistently localized to layers 15–31, with a clearly discontinuous interval mainly induced by the sudden drop of κ_ℓ. Editing within this gap tends to degrade editing performance, as confirmed by the layer-wise ablation reported in Appendix I.

Other backbones. The localization patterns of the remaining backbones share the same structural trend observed in BLIP2-OPT-2.7B: *(i)* the selected layers always form a discontinuous set, in which the last segment is continuous and spans a substantial range, whereas the first segment typically covers only a few layers; and *(ii)* the precise distribution of these discontinuities is architecture-dependent. For instance, the localized interval of LLaVA-1.5-7B (Fig. 14 and Fig. 15) is shifted earlier relative to BLIP2-OPT-2.7B and split into two segments, MiniGPT4-Vicuna-7B (Fig. 16 and Fig. 17) is partitioned into three, and the Qwen family (Fig. 18, Fig. 19, Fig. 20, and Fig. 21) exhibits a continuous interval near the terminal layers together with a near-layer-wise alternation at the earliest layers.

Discussion. Overall, the layer set selected by (C1)–(C3) is nearly invariant to the choice of edit instances within a given backbone. We therefore conjecture that these three conditions encode a structural property of how VLLMs internally route and re-integrate multimodal information, providing a principled localization criterion for locate-then-edit knowledge editing.

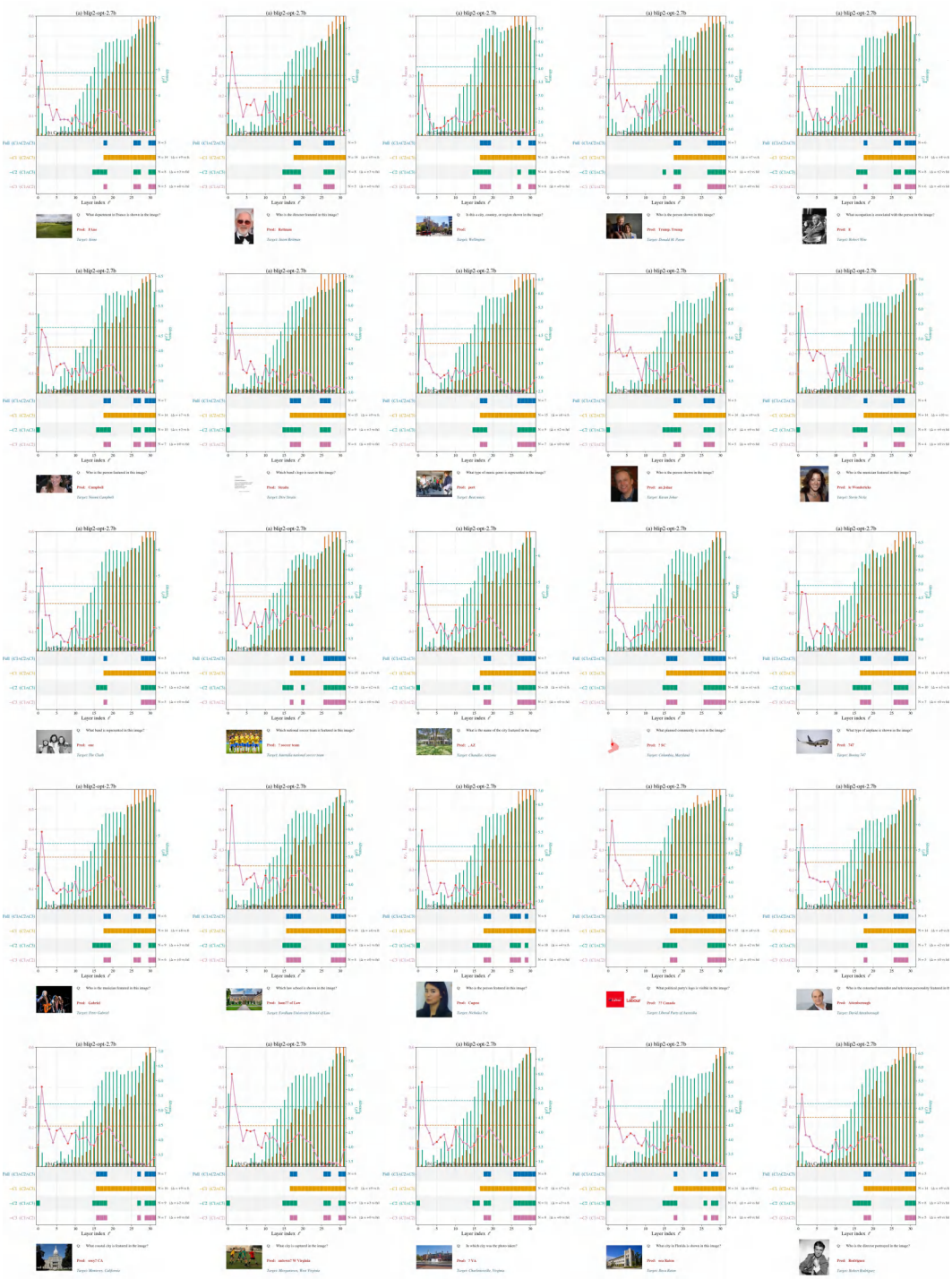

Figure 12: Visualization results with BLIP2-OPT-2.7B on VLKEB dataset (a).

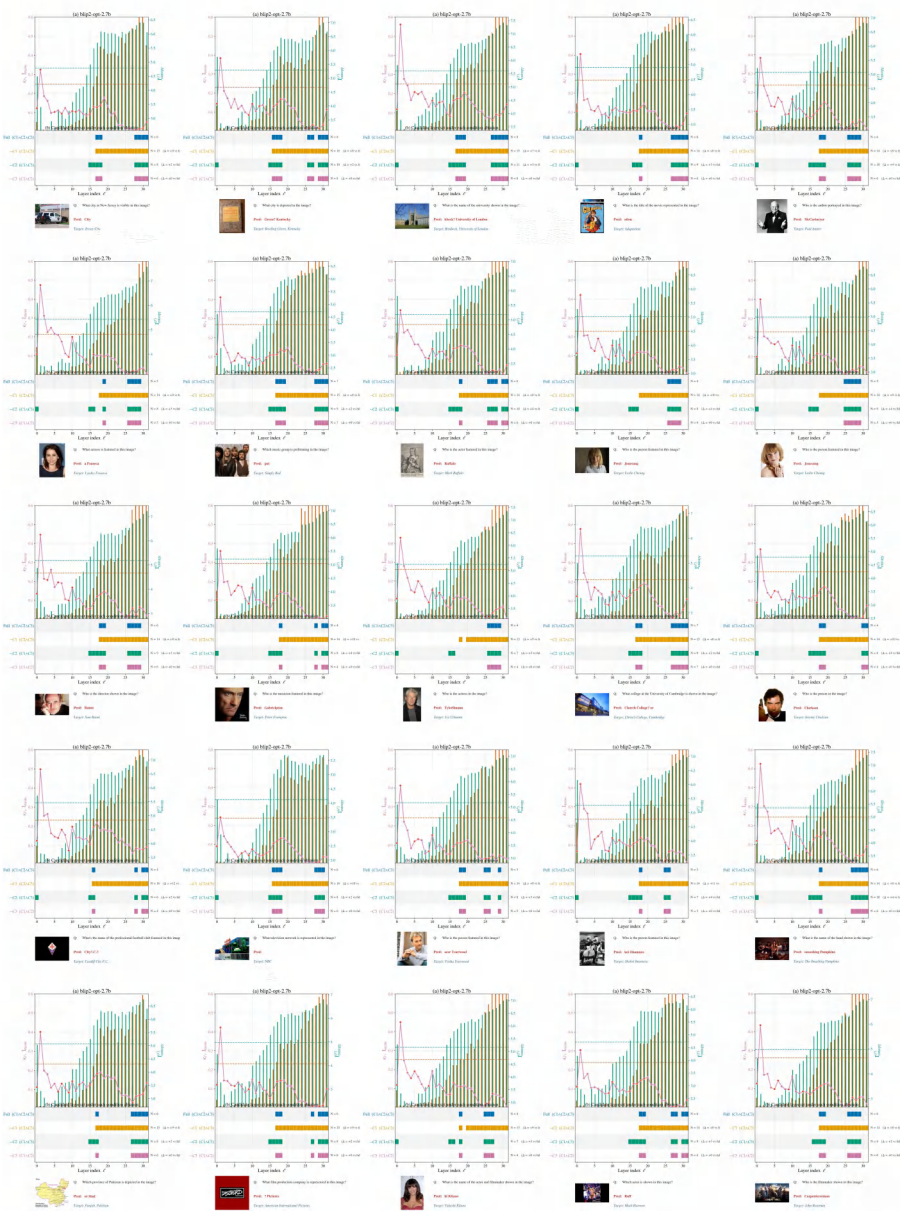

Figure 13: Visualization results with BLIP2-OPT-2.7B on VLKEB dataset (b).

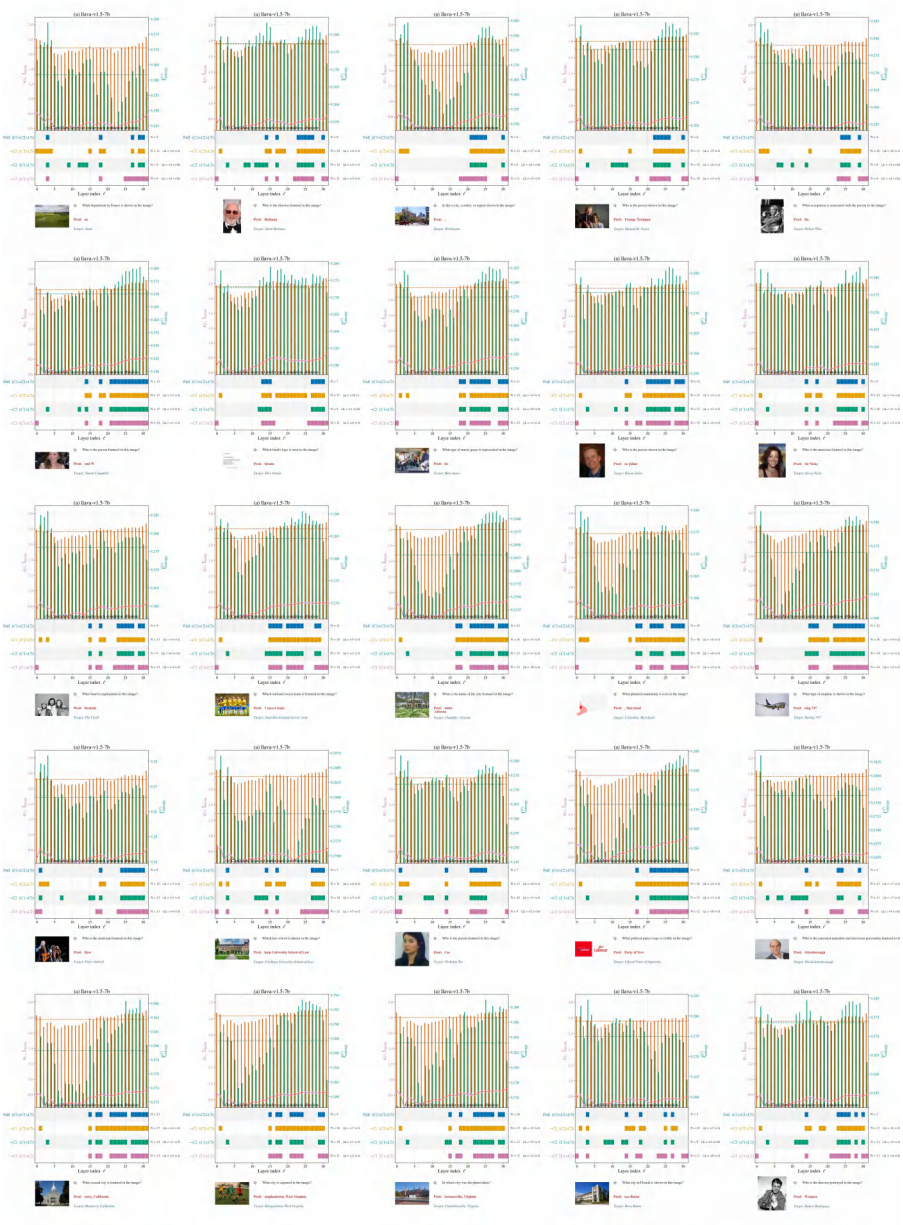

Figure 14: Visualization results with LLaVA-v1.5-7B on VLKEB dataset (a).

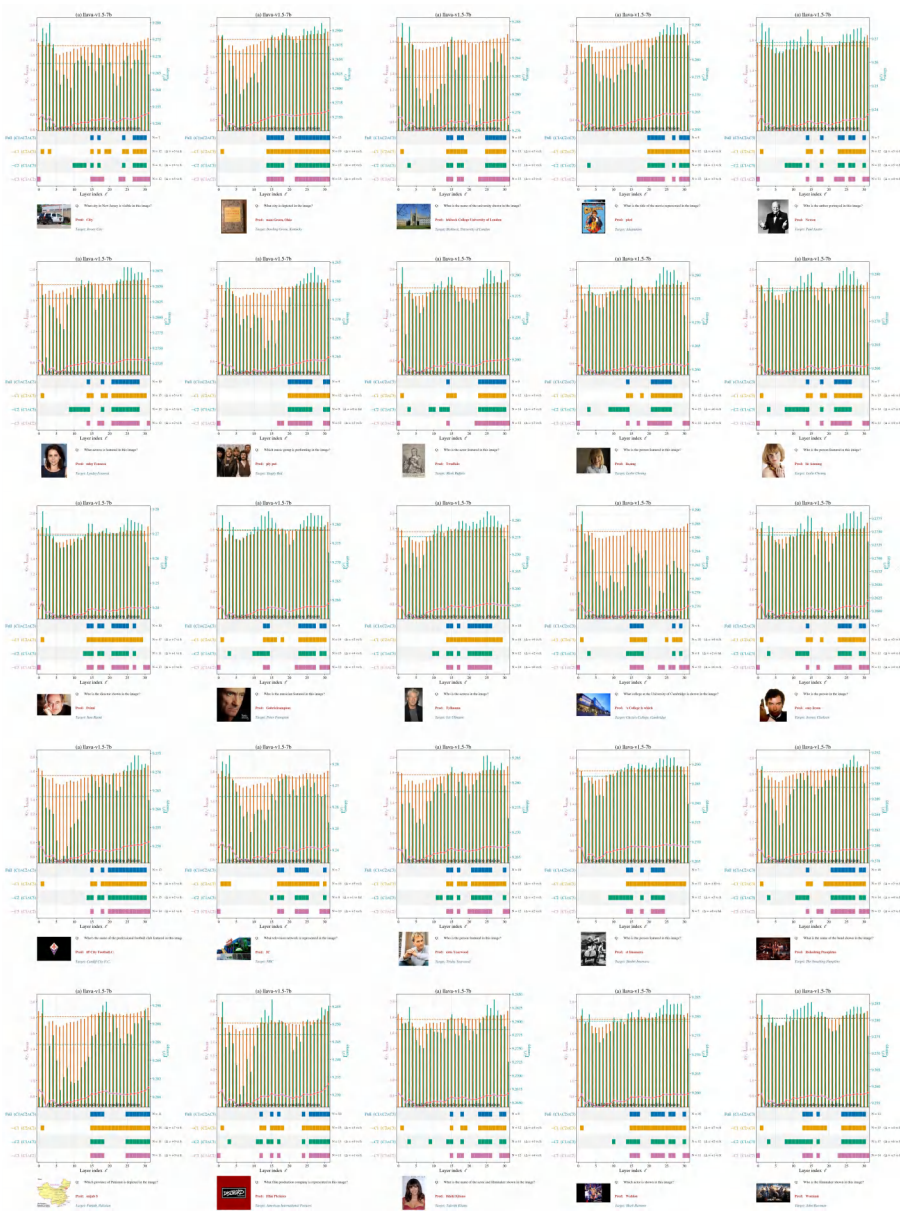

Figure 15: Visualization results with LLaVA-v1.5-7B on VLKEB dataset (b).

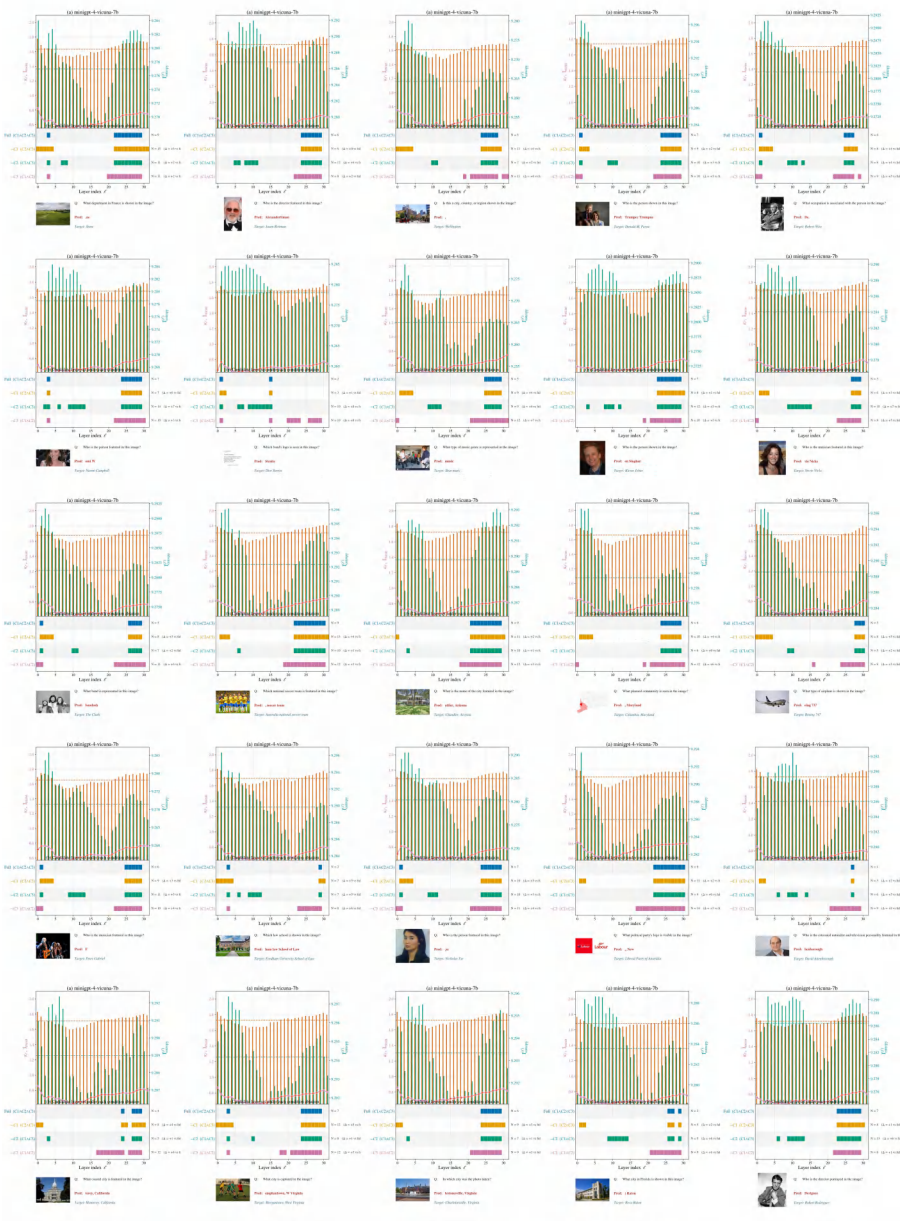

Figure 16: Visualization results with MiniGPT4-Vicuna-7B on VLKEB dataset (a).

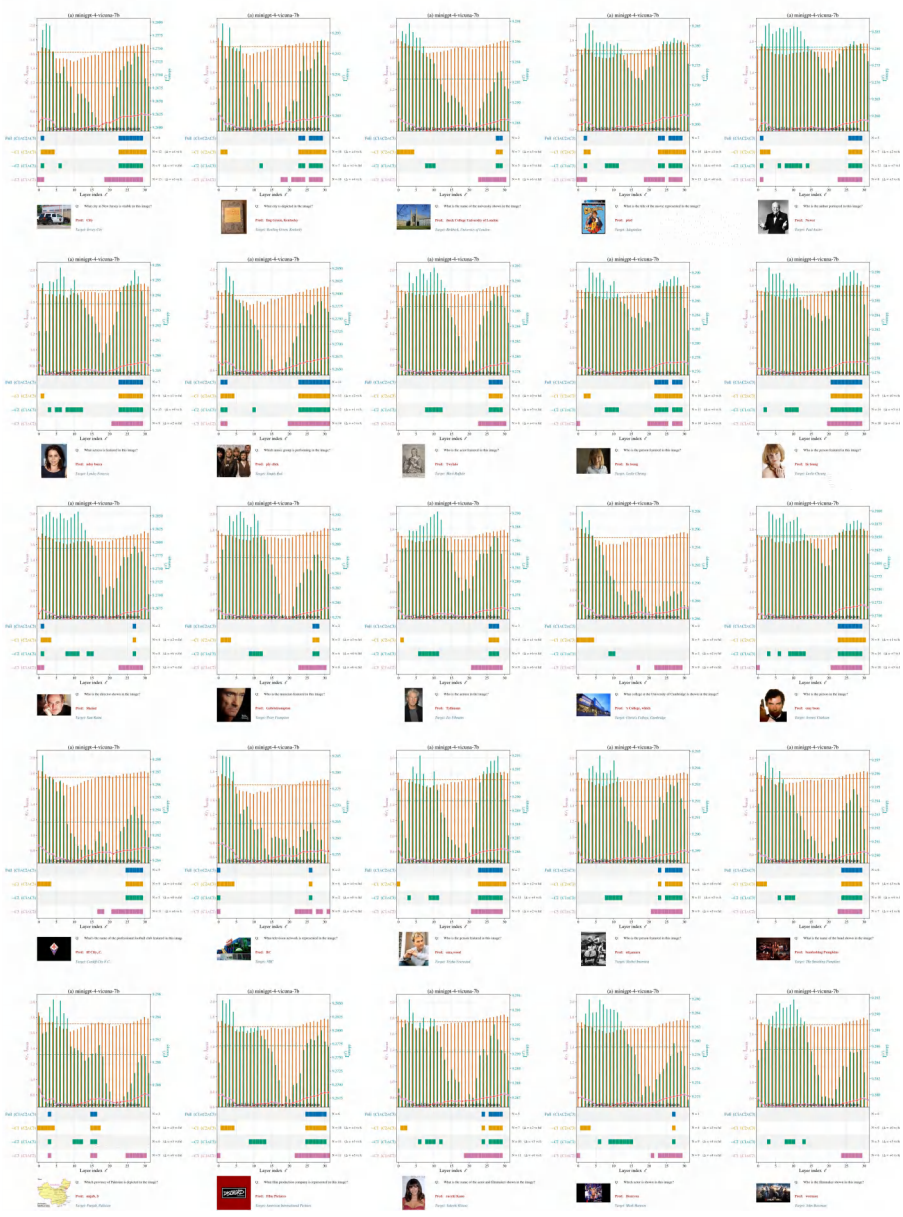

Figure 17: Visualization results with MiniGPT4-Vicuna-7B on VLKEB dataset (b).

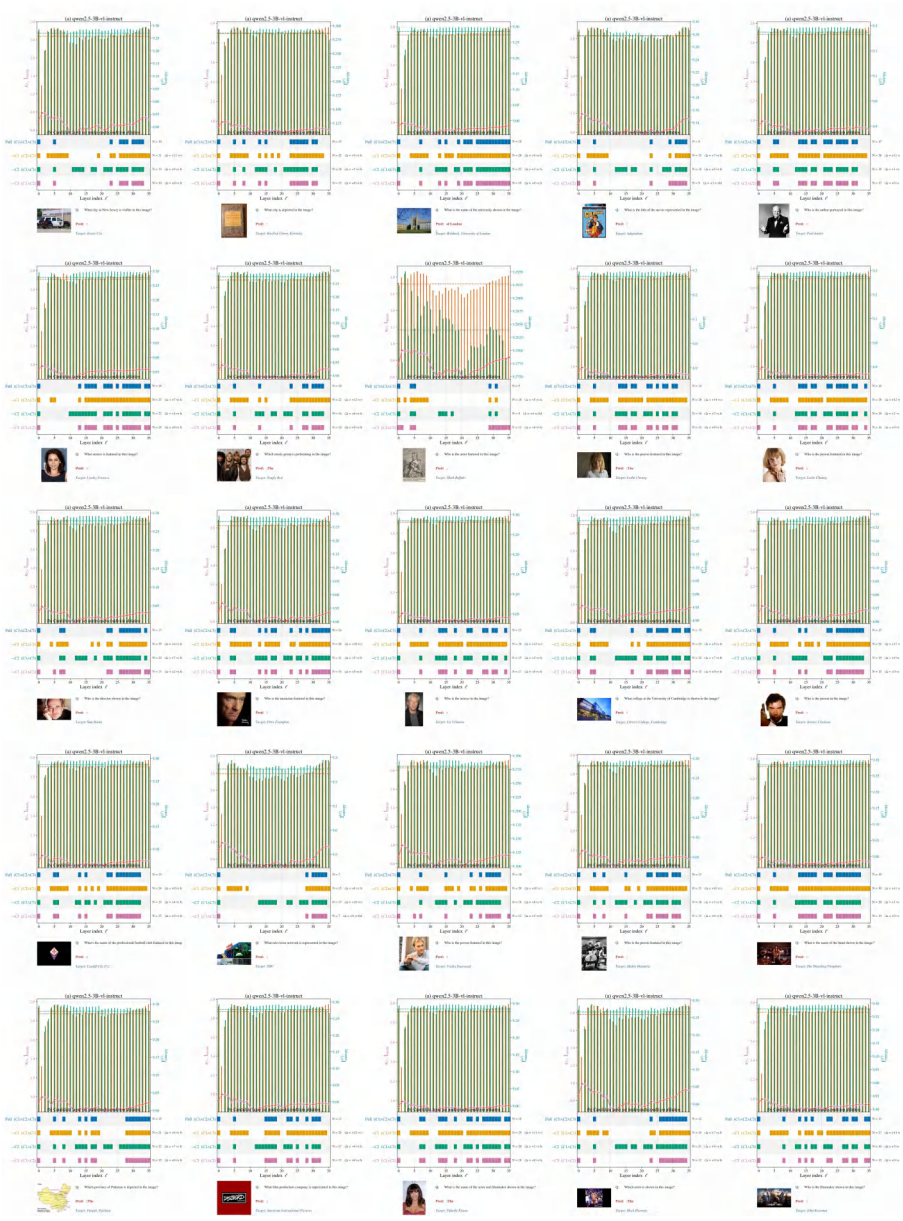

Figure 18: Visualization results with Qwen2.5-VL-3B-Instruct on VLKEB dataset (a).

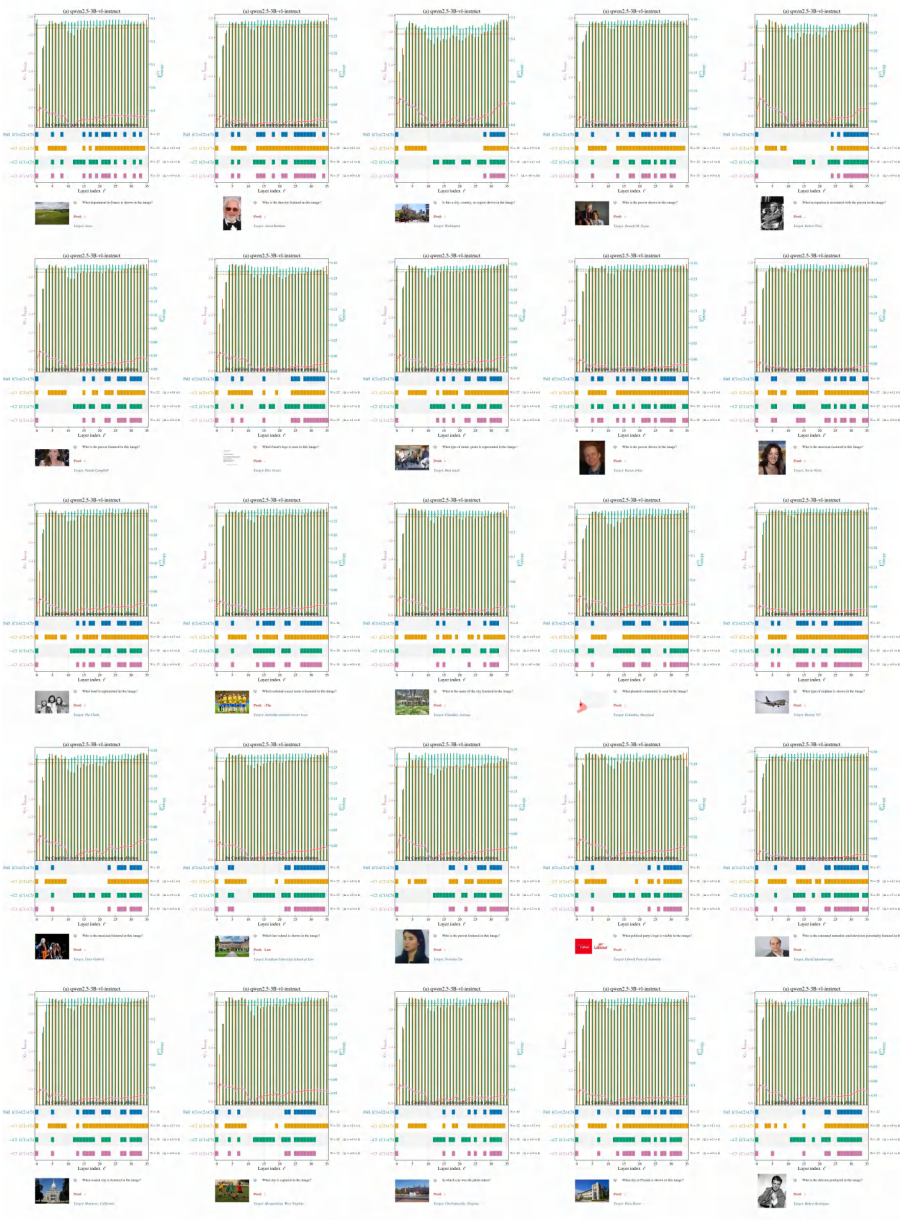

Figure 19: Visualization results with Qwen2.5-VL-3B-Instruct on VLKEB dataset (b).

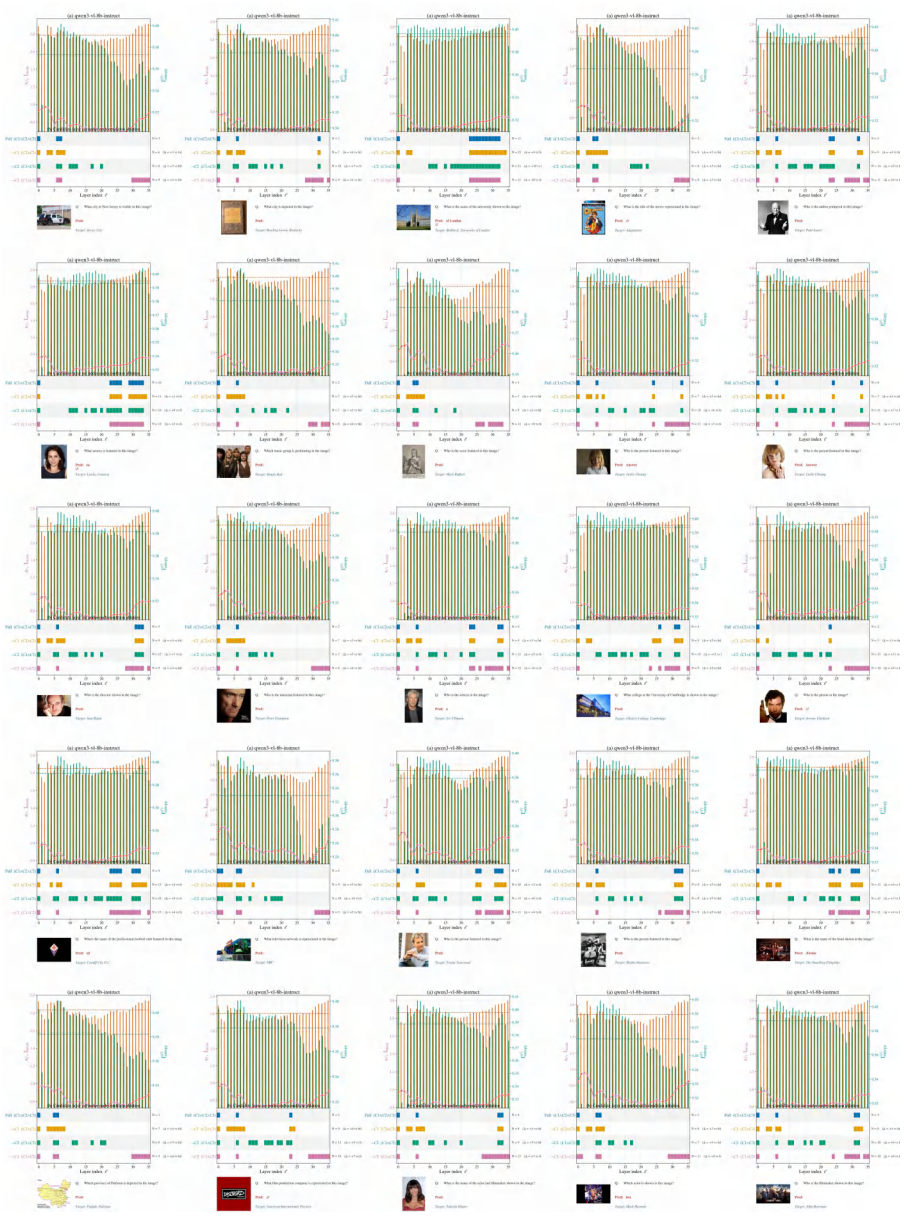

Figure 20: Visualization results with Qwen3-VL-8B-Instruct on VLKEB dataset (a).

H Vision Dominance Threshold Upper Bound Estimation

As mentioned in Eq. 8, we set $\rho_j^{(\ell),\text{vis}} > \tau_{\text{vis}}$ to filter neurons with excessive visual dominance. We assert that τ_{vis} should be lower than $1 + \frac{N_t}{N_v}$. The following derivation provides the proof.

$$\begin{aligned}\rho_j^{(\ell),\text{vis}} = \frac{\bar{a}_j^{(\ell),\mathcal{P}_{\text{vis}}}}{|\bar{a}_j^{(\ell),\mathcal{P}_{\text{all}}}| + \epsilon} < \frac{\bar{a}_j^{(\ell),\mathcal{P}_{\text{vis}}}}{\bar{a}_j^{(\ell),\mathcal{P}_{\text{all}}}} = \frac{\frac{1}{|\mathcal{P}_{\text{vis}}|}\sum_{i\in\mathcal{P}_{\text{vis}}} \bar{a}_{i,j}^{(\ell)}}{\frac{1}{|\mathcal{P}_{\text{all}}|}\sum_{i\in\mathcal{P}_{\text{all}}} \bar{a}_{i,j}^{(\ell)}} \leq \frac{|\mathcal{P}_{\text{all}}|}{|\mathcal{P}_{\text{vis}}|} \\ \leq \frac{N_v + N_t}{N_v} = 1 + \frac{N_t}{N_v}.\end{aligned} \tag{39}$$

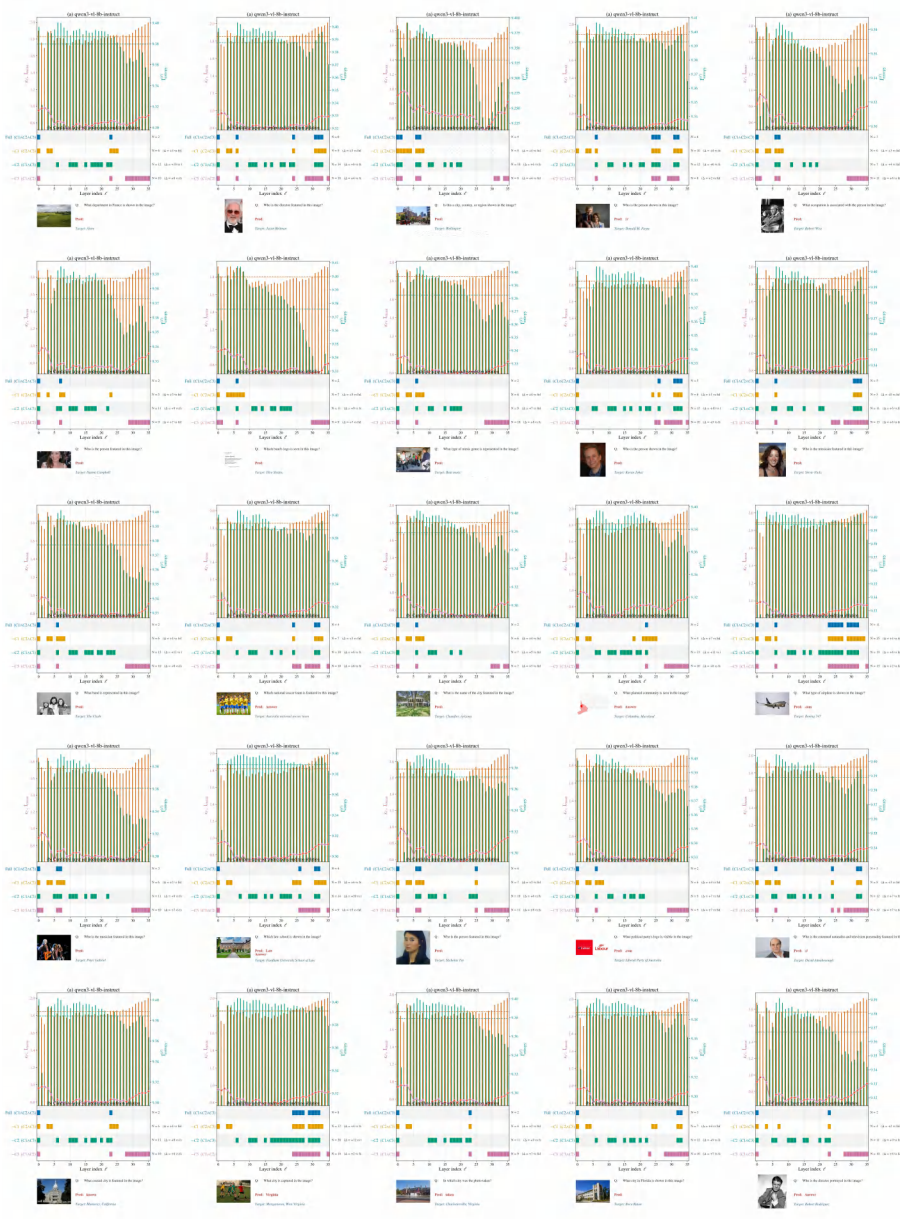

Figure 21: Visualization results with Qwen3-VL-8B-Instruct on VLKEB dataset (B).

Note that each $\bar{a}_{i,j}^{(\ell)} > 0$.

I MORE ABLATION STUDIES

In this section, we report the experimental results mentioned in Appendix G (see Tab. 6), and provide a more detailed analysis of the ablation studies discussed in the main body. Furthermore, Tab. 7 displays the combinations of localization criteria (C1)–(C3) under a lifelong setting, e.g., # Edits = 10. As discussed in the main body, we observe that the decline in editing performance becomes more severe as additional criteria are excluded. The combinations labeled (a)–(g) in the main body correspond to the exact layer selections shown in Tab. 5.

C1	C2	C3	Notation	$\mathcal{L}$
✓	✓	✓	**Full** (C1+C2+C3)	{17, 18, 19, 26, 27, 28, 29, 30, 31}
○	✓	✓	*(a)*	{17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31}
✓	○	✓	*(b)*	{0, 16, 17, 18, 19, 26, 27, 29, 30, 31}
✓	✓	○	*(c)*	{15, 16, 17, 18, 19, 26, 27, 28, 29, 30, 31}
○	○	✓	*(d)*	{0, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31}
○	✓	○	*(e)*	{17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31}
✓	○	○	*(f)*	{0, 1, 4, 5, 6, 9, 10, 16, 17, 18, 19, 26, 27, 29, 30, 31}
○	○	○	*(g)*	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31}

Table 5: All 2^3=8 combinations of criteria C1, C2, and C3 for localization. ✓: criteria enabled; ○: criteria disabled. The rightmost column is reserved for labels notation.

	Cond	**Rel**	**T-Gen**	**M-Gen**	**T-Loc**	**M-Loc**	**Avg**
$\mathcal{L}^{\text{extra}}$+**Full***		92.06	74.96	89.52	98.59	79.30	86.88
Full*		**96.28**	**85.48**	**94.91**	**99.58**	**83.96**	**92.04**

Table 6: As illustrated in Fig. 12 (row 4, col 2), we add $\mathcal{L}^{\text{extra}} = \{20, 21, \ldots, 26\}$ to the candidate layer set $\mathcal{L}_{\text{BLIP}}$ and conduct experiments with BLIP2-OPT-2.7B on the EVQA dataset.

Cond	**Rel**	**T-Gen**	**M-Gen**	**T-Loc**	**M-Loc**	**Avg**
(a)	37.91	34.40	33.90	80.21	54.63	48.21
(b)	37.91	34.40	33.90	89.21	60.03	51.09
(c)	40.33	35.18	36.38	91.05	87.67	58.12
(d)	79.49	68.87	72.42	95.90	28.98	69.13
(e)	78.65	63.61	72.12	90.23	28.15	66.55
(f)	70.01	67.06	70.28	95.24	30.77	66.67
(g)	64.40	57.54	59.68	90.70	26.35	59.73
Full	**85.95**	**76.89**	**76.11**	91.94	**61.60**	**78.50**

Table 7: Layer ablation on the EVQA dataset with # Edits - 10.