Group Relative Attention Guidance for Image Editing

Supplementary Material

Contents

A. Motivation Clarification.

Figure. 4 and Figure. 5 provide the motivation for grouping and intra-group reweighting in Algorithm 1, respectively. As shown in Figure. 4, combined with the properties of RoPE encoding along the dimension axis, this observation suggests that textual and visual features tend to organize into distinct clusters in a shared embedding space. This observation motivates the grouping strategy in Sec. 5. We further compute the ℓ_2 norm of the mean vector within each group and observe that it remains stable across different inputs (see Figure. 5, Sec. G, and Table S1). In other words, input variations manifest as deviations of group-wise tokens from their corresponding mean vectors. These analyses motivate us to control the editing outcomes by adjusting the variation components.

B. Pytorch Implementation of GRAG

The proposed **Group Relative Attention Guidance (GRAG)** can be seamlessly integrated into existing DiT-based image editing models with only a few lines of code modification. Below, we provide an example implementation of GRAG based on a typical MM-Attention block from the `Diffusers` library in PyTorch.

Listing 1. Implementation Code of GRAG

```
# Apply RoPE
if image_rotary_emb is not None:
    img_freqs, txt_freqs = image_rotary_emb
    img_query = apply_rotary_emb_qwen(img_query, img_freqs, use_real=False)
    img_key = apply_rotary_emb_qwen(img_key, img_freqs, use_real=False)
    txt_query = apply_rotary_emb_qwen(txt_query, txt_freqs, use_real=False)
    txt_key = apply_rotary_emb_qwen(txt_key, txt_freqs, use_real=False)

# Apply GRAG scaling
s_idx, e_idx, bias_scale, delta_scale = 4096, 8192, 1.0, 1.05
group_bias = img_key[:, s_idx:e_idx, :, :].mean(dim=1)
img_key[:, s_idx:e_idx, :, :]  = bias_scale * group_bias +
                                 delta_scale * (img_key[:, s_idx:e_idx, :, :]  - group_bias)

# Joint attention computation
joint_query = torch.cat([txt_query, img_query], dim=1)
joint_key = torch.cat([txt_key, img_key], dim=1)
joint_value = torch.cat([txt_value, img_value], dim=1)

joint_hidden_states = dispatch_attention_fn(
    joint_query,
    joint_key,
    joint_value,
    attn_mask=attention_mask,
    dropout_p=0.0,
    is_causal=False,
    backend=self._attention_backend,
)
```

C. Theoretical Analysis of Group Relative Attention Guidance

C.1. Toy Experiment of the GRAG Theoretical Analysis

To further validate the theoretical analysis presented in the main paper and Section 5, we design a controlled toy experiment that isolates the effect of Group Relative Attention Guidance (GRAG) on attention score modulation. The aim is to empirically verify how adjusting the deviation scaling factor δ and the bias scaling factor λ influences the distribution of attention across token groups.

Experimental Setup. We simplify the MM-Attention mechanism in a minimal setting consisting of a single query $q = 1$ and three key tokens.

$$A(q, k^i) = \frac{e^{\langle q, k^i \rangle}}{e^{\langle q, k_s^1 \rangle} + e^{\langle q, k_s^2 \rangle} + e^{\langle q, k_t \rangle}}, \tag{S1}$$

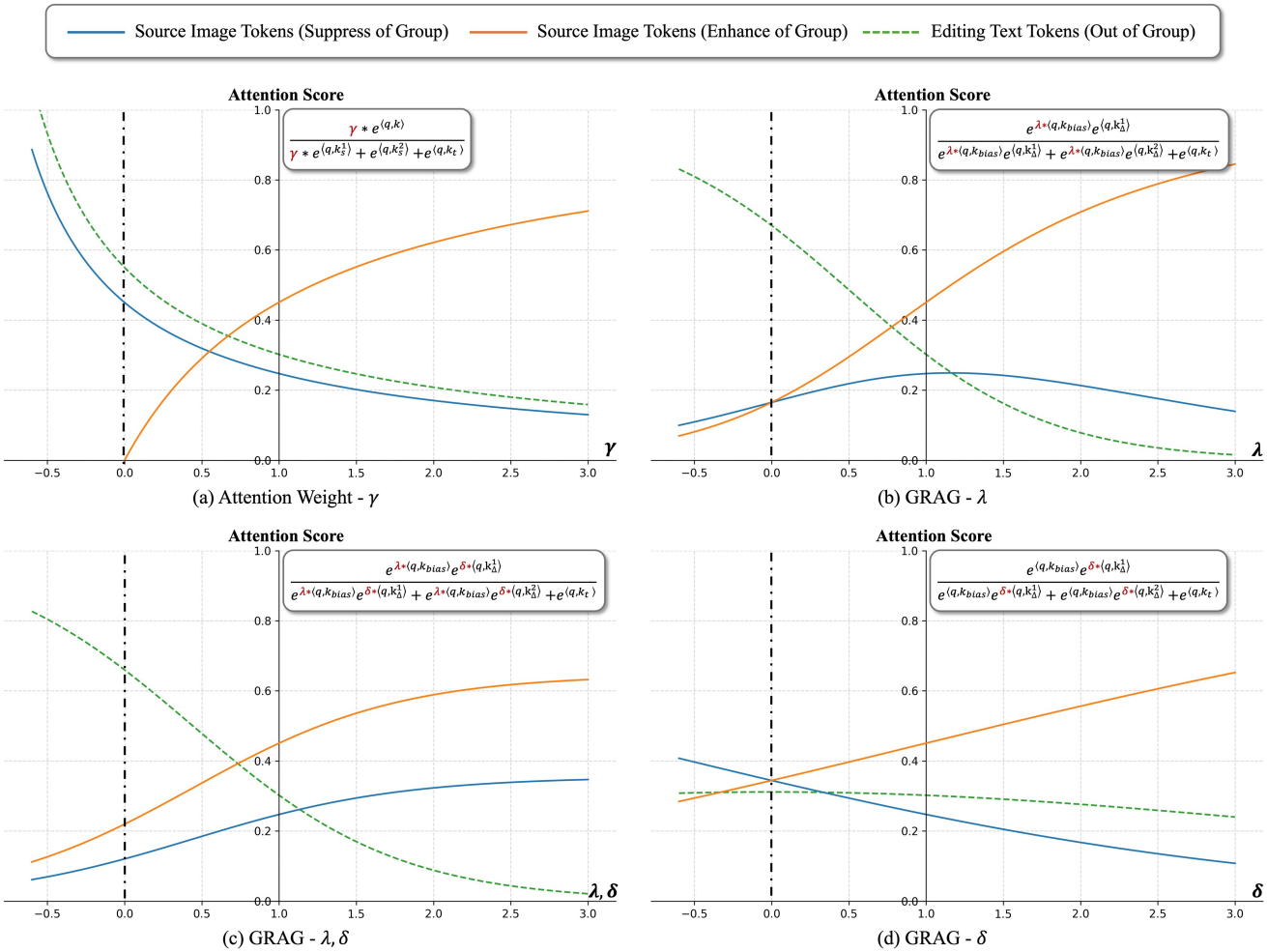
where $k_s^1 = 1.9$ and $k_s^2 = 1.2$ denote two source-image tokens and $k_t = 3.4$ denotes the editing-text token. Following the setup in Section 5 of the main paper, two tokens form the *source image token group*, while the remaining token represents the *editing text token*. The raw inner-product responses $\langle q, k^i \rangle$ of these tokens serve as the unmodulated attention logits. This abstraction preserves the structural essence of text–image cross-attention while allowing us to precisely examine how GRAG affects token-wise attention allocation. Ideally, when increasing editing strength toward higher image–content consistency, the model should *maintain responsiveness to the editing instruction* while *smoothly increasing* the contribution of selected source-image tokens.

We compare four representative modulation strategies:

- Attention Weight - γ: Directly scaling attention scores after the Softmax layer.
- GRAG - λ: Modulating only the bias component while holding the deviation term fixed.
- GRAG - λ, δ: Jointly scaling both the bias component and token-level deviations.
- GRAG - δ: Modulating only the deviation component while keeping the bias fixed.

Experimental Results. Figure S1 illustrates this desired behavior using a toy example. The horizontal axis denotes the respective modulation parameter, and the vertical axis shows the post-Softmax attention scores of three token groups: suppressed source-image tokens, enhanced source-image tokens, and editing-text tokens. As seen in subfigures (a)–(c), increasing γ or λ, or jointly tuning λ and δ, rapidly suppresses attention to the editing-text tokens, causing the editing instruction to collapse and resulting in artifacts or editing failure despite increased attention to the image tokens.

In contrast, deviation-only modulation via GRAG-δ preserves stable attention on editing-text tokens while smoothly adjusting the relative emphasis among source-image tokens. This yields the most continuous and controllable editing-strength transition, free of undesirable discontinuities or instruction loss. Overall, GRAG-δ achieves the desired balance between instruction following and image consistency, providing the most reliable and precise editing-strength control among all tested strategies.



(a) Attention Weight - γ (b) GRAG - λ

(c) GRAG - λ, δ (d) GRAG - δ

Figure S1. **Toy Experiment:** Effects of different attention modulation strategies on the attention scores within MM-Attention. GRAG-δ enhances or suppresses image tokens while preserving responsiveness to the editing instruction, enabling continuous and precise control over editing strength. The *left vertical line* marks the lower modification boundary, and the *y-axis* denotes the baseline without any modulation.

C.2. Relationship Between Attention Entropy and GRAG

To quantify how GRAG influences the distribution of attention inside MM-Attention, we measure the *attention entropy* of each attention map. Given an attention distribution $A^{(l)} \in \mathbb{R}^N$ from layer l, its entropy is defined as:

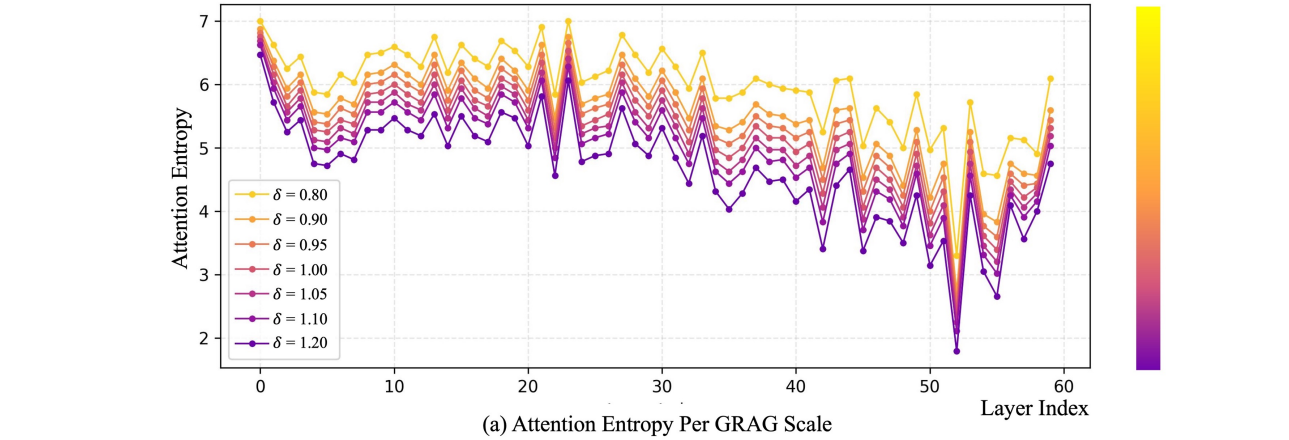
$$H(A^{(l)}) = -\sum_{i=1}^{N} A_i^{(l)} \log A_i^{(l)}, \tag{S2}$$

where lower entropy indicates a more concentrated attention pattern and stronger focus on specific tokens.

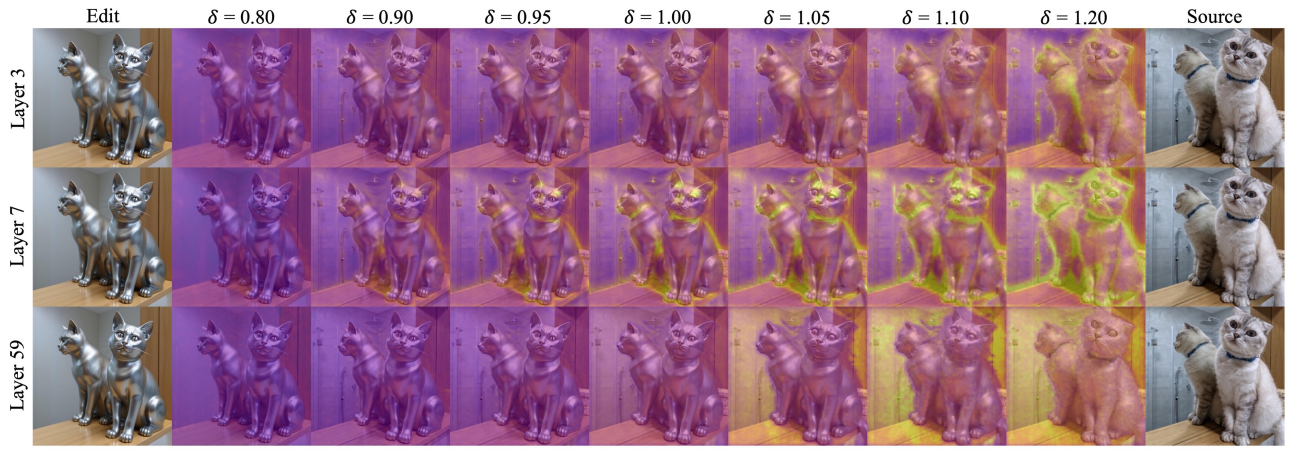
We compute attention entropy for all layers of the Qwen-Image-Edit model under different GRAG values. Specifically, for each GRAG scale, we extract the attention maps associated with the source-image tokens and compute their entropies using Equation (S2). The aggregated results are shown in Figure S2(a). As the GRAG value increases, the attention entropy in every layer decreases monotonically, indicating that GRAG strengthens the model's focus on conditioning information and thereby increases the effective editing strength.

To provide a more intuitive interpretation, we visualize the attention maps corresponding to the same settings. As shown in Figure S2(b), increasing the GRAG value causes the query's attention to progressively concentrate on the original cat contours. This focused attention corresponds to stronger preservation of source-image structure, leading to improved consistency between the edited output and the reference image.

These results collectively demonstrate that GRAG modulates editing strength by adjusting the concentration of cross-attention, aligning the empirical observations with the theoretical analysis presented in the main paper.



(a) Attention Entropy Per GRAG Scale



(b) Attention Map Per GRAG Scale

Figure S2. Relationship between attention entropy and GRAG. (a) Increasing the GRAG value on source image tokens leads to a monotonic decrease in attention entropy across layers. (b) The query's attention over source image tokens becomes progressively more concentrated as the GRAG value increases.

C.3. GRAG of Source Image Tokens

To further understand the distinct roles of the bias and variation components in GRAG, we separately perform continuous adjustments on the text-embedding scaling factors λ (bias scaling) and δ (variation scaling).

$$\hat{k}_{\mathrm{s}}^{i} = \lambda \cdot k_{\mathrm{bias}} + \delta \cdot \left(k_{\mathrm{s}}^{i} - k_{\mathrm{bias}}\right) \tag{S3}$$

The results are shown in Figure. S3. When only the bias component is modulated, the overall editing effect remains relatively stable, yet noticeable changes in image quality occur. In contrast, when only the variation component is adjusted, the semantic content of the edited image changes rapidly with increasing editing strength, while the overall image quality remains largely consistent. These observations support our interpretation that the bias component corresponds to the model's intrinsic editing behavior, whereas the variation component encodes the actual content-specific editing signals. In particular, when $\delta = 0$, scaling only the bias term with λ still changes the generated image, but the change is largely independent of the specific editing condition. This supports our interpretation that the bias component captures a content-agnostic editing action, while the deviation term carries the condition-specific signal.

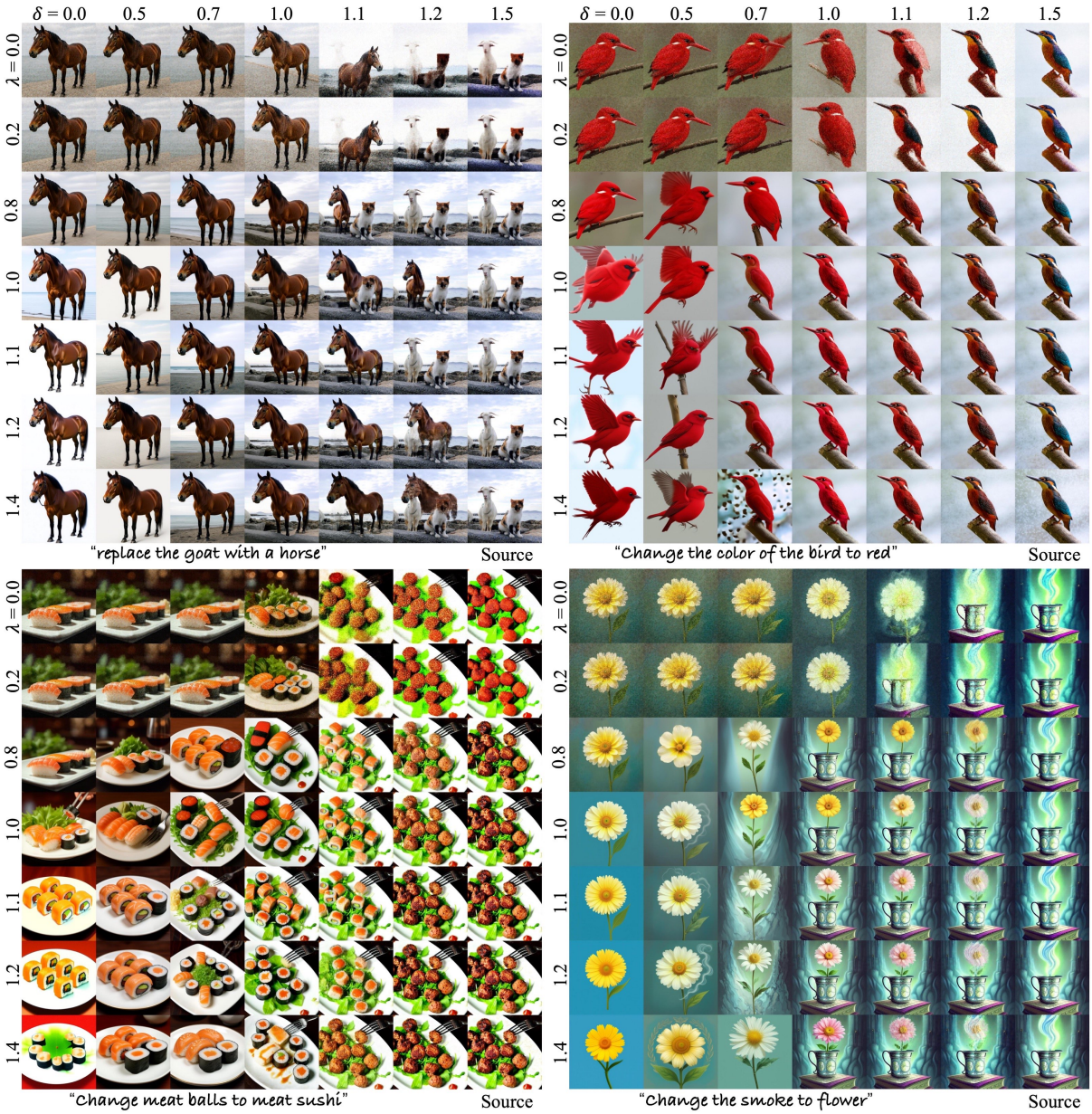


Figure S3. Effects of the bias and variation components in image-key embeddings.

C.4. GRAG of Editing Text Tokens

We conduct the same decomposition-based analysis on the text tokens to examine the roles of the bias and variation components, with results shown in Figure. S4. Similar to the observations for image tokens, the bias component governs the model's inherent editing behavior, while the variation component drives the content-specific semantic changes. However, applying GRAG to text tokens reveals a noticeably stronger degree of semantic-level editing control, indicating that modulation of text embeddings provides a more direct and expressive pathway for manipulating editing intensity. The same trend also appears for text tokens: when $\delta = 0$, adjusting only the bias term produces changes that are weakly coupled to the instruction content, again suggesting that the deviation term is the main carrier of instruction-specific semantics.

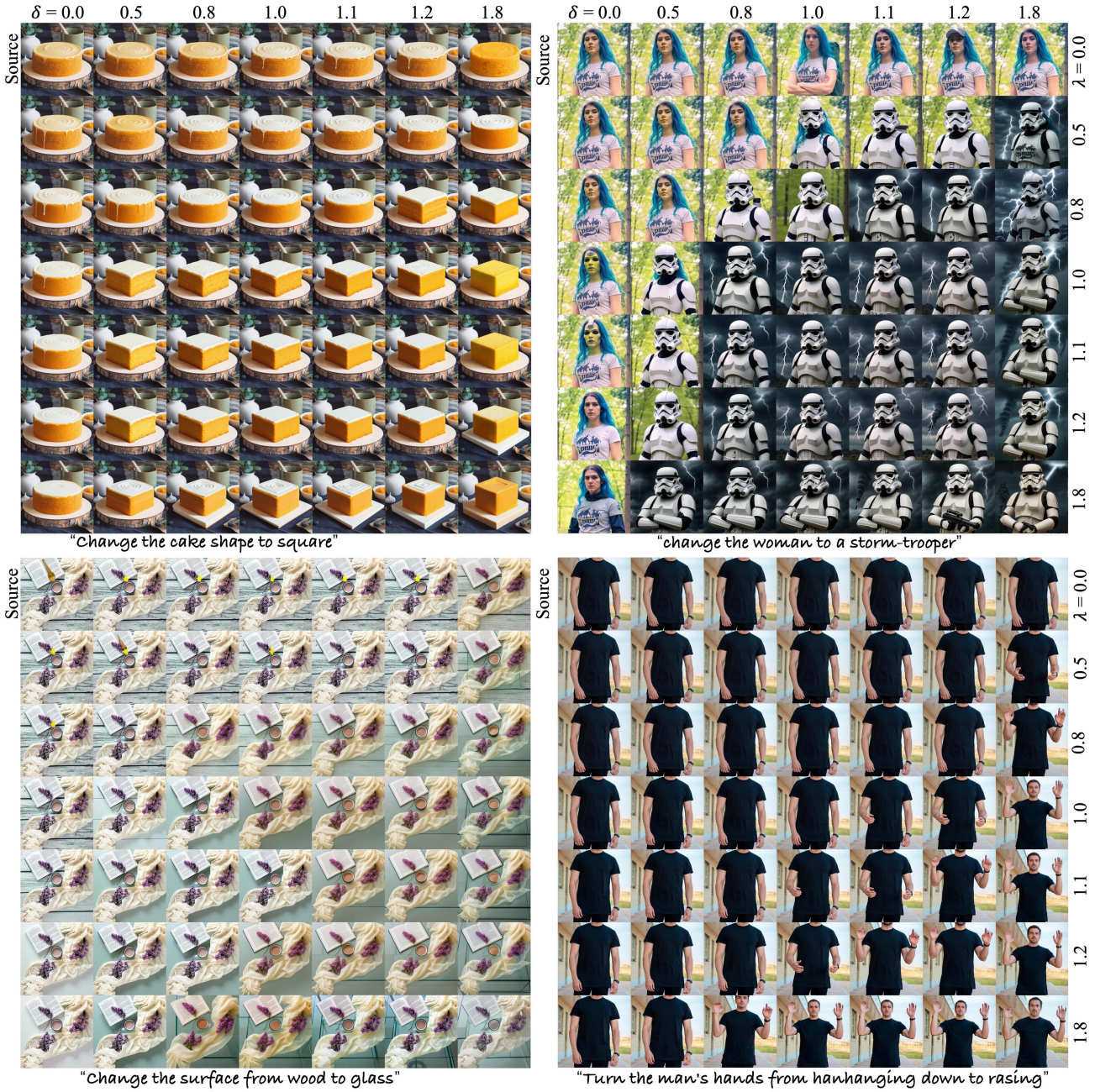


Figure S4. Effects of the bias and variation components in text-key embeddings.

C.5. Bias-Vector Generality Across Samples

To verify that the bias vector is not an artifact of using batch size 1, we compute the ℓ_2 norm of the group mean vector on 1, 10, 100, 1,000, and 10,000 image-editing pairs. As shown in Table S1, both the text and image bias vectors remain highly stable as the sample size grows, with only minor variance changes. This confirms that the bias vector is a robust property of the learned embedding space rather than a sample-specific observation.

Samples	**1**	**10**	**100**	**1000**	**10000**
Text-bias-vector	10.08±0	9.91±0.32	10.03±0.19	10.02±0.20	10.01±0.20
Img-bias-vector	7.20±0	7.39±0.19	7.37±0.18	7.38±0.20	7.37±0.20

Table S1. ℓ_2 norm of the bias vector across different sample sizes.

D. Limitation & Discussion

We observe that the effectiveness of GRAG is inherently limited when applied to training-free image editing methods. As illustrated in Figure. S5, training-based TI2I models possess a unified architecture in which the editing instruction and source-image information interact within the same MM-Attention layers, yielding better compatibility with GRAG. In contrast, training-free approaches are built upon T2I models that are not originally designed for editing; they rely on additional inversion procedures and attention injection to approximate the interaction between edited and source-image features. This structural mismatch constrains the impact of GRAG and lead to degraded image quality or ineffective edits (Figure. S6).

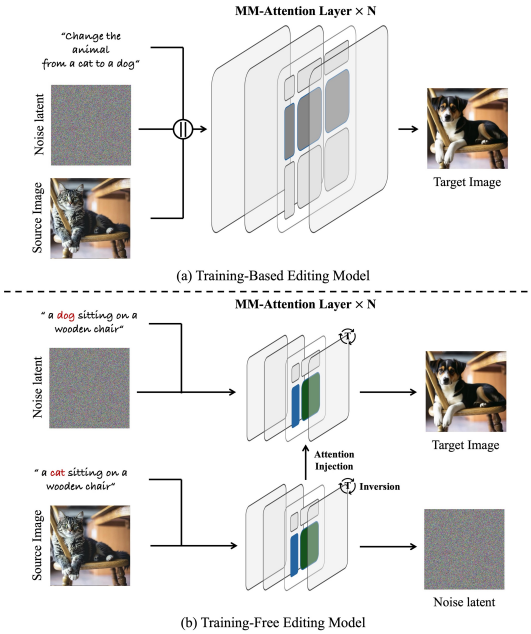


Figure S5. Structural differences between training-based and training-free editing methods.

Figure S6. Failure cases of applying GRAG to training-free editing methods.

E. Quantitative Comparison of Guidance Strategies

Corresponding to Figure 10 in the main text, Table S2 reports the full PIE metrics under different settings of CFG, attention reweighting, attention gating, and GRAG.

Method	LPIPS ↓	SSIM↑	Cons ↑	PF ↑	EditScore ↑
CFG = 5.00	0.3381	0.8548	8.3989	8.4640	7.1857
CFG = 4.00	0.3428	0.8506	8.5211	8.4806	7.2576
CFG = 3.00	0.3312	0.8659	8.6251	8.3954	7.2761
CFG = 2.00	0.3297	0.8709	8.6669	8.2566	7.2247
CFG = 1.00	0.3306	0.8734	8.7686	7.6760	6.8294
Attn-weight = 0.8	0.3223	0.8940	8.5846	8.4503	7.2808
Attn-weight = 1.0	0.3428	0.8506	8.5211	8.4806	7.2576
Attn-weight = 7.0	0.3055	0.9261	8.8301	8.2159	7.2214
Attn-weight = 10.0	0.2916	0.9484	9.1130	6.7797	5.9555
Attn-weight = 16.0	0.2825	0.9560	9.2361	4.5230	3.9953
Attn-gate = 1.0	0.3428	0.8506	8.5211	8.4806	7.2576
Attn-gate = 0.9	0.3094	0.9031	8.6460	8.1206	7.4004
Attn-gate = 0.7	0.3103	0.9033	8.6655	7.9611	7.2531
Attn-gate = 0.3	0.3296	0.8673	8.1051	7.7184	4.6505
$\lambda = 0.95, \delta = 1.00$	0.3316	0.8660	8.5286	8.4886	7.2725
$\lambda = 1.00, \delta = 1.00$	0.3428	0.8506	8.5211	8.4806	7.2576
$\lambda = 1.05, \delta = 1.00$	0.3123	0.9156	8.5914	8.3977	7.2990
$\lambda = 1.10, \delta = 1.00$	0.3194	0.9034	8.3291	7.9251	7.1992
$\lambda = 1.15, \delta = 1.00$	0.3307	0.8865	8.3720	8.3269	7.1863
$\lambda = 1.00, \delta = 0.95$	0.3508	0.8344	8.2543	8.4394	7.1991
$\lambda = 1.00, \delta = 1.00$	0.3428	0.8506	8.5211	8.4806	7.2576
$\lambda = 1.00, \delta = 1.05$	0.3188	0.8855	8.7549	8.3651	7.2679
$\lambda = 1.00, \delta = 1.10$	0.3034	0.9206	8.9537	7.9611	6.9872
$\lambda = 1.00, \delta = 1.15$	0.2907	0.9408	9.1731	6.9291	6.1730
$\lambda = 0.95, \delta = 0.95$	0.3454	0.8434	8.3560	8.3394	7.2162
$\lambda = 1.00, \delta = 1.00$	0.3428	0.8506	8.5211	8.4806	7.2576
$\lambda = 1.05, \delta = 1.05$	0.3042	0.9263	8.9440	8.3303	7.3245
$\lambda = 1.10, \delta = 1.10$	0.2971	0.9391	9.0234	7.9674	7.0243
$\lambda = 1.15, \delta = 1.15$	0.2885	0.9448	9.1051	6.6091	5.9955

Table S2. Continuity and effectiveness analysis of different editing strength control methods. Appropriate parameters are selected for each method within its effective working range.

F. Additional Qualitative Results

(a) Kontext-Dev

(b) Qwen-Edit

(c) Step1X-Edit

Figure S7. Additional qualitative results on existing image editing models.

Figure S8. Additional comparison results with existing editing-strength control methods.

G. Additional Feature Visualization

We provide additional Kontext and Qwen-Edit model feature distribution statistics corresponding to Figures 2, 4, and 5 in the main paper. Consistent with the experiments presented in the main text, we analyze different image editing samples (IDs) across various denoising steps and model layers to examine the correlation between feature distributions and these three factors. Figure S9, Figure S15 presents direct visualizations of the feature distributions, where the *TokenNumber* dimension is downsampled by a factor of 4 and the *Dim* dimension by a factor of 2. Figure S10, Figure S16 shows aggregating different tokens along the sequence dimension. Figures S11–S14, Figures S17–S20 illustrate the mean and variance of token-wise feature distributions across different attention heads, corresponding to the different embedding features.

G.1. Kontext Embedding Visualization

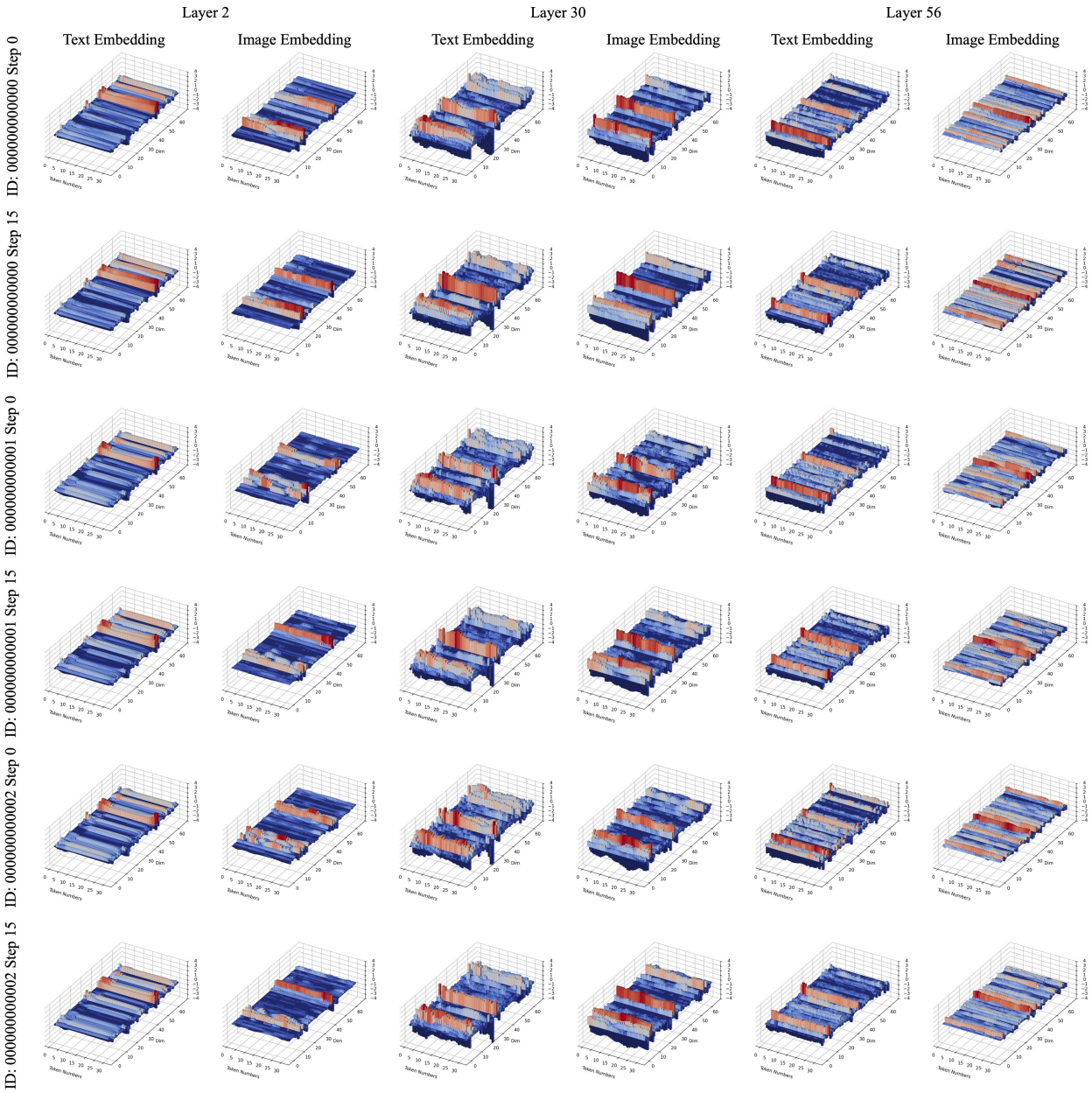


Figure S9. Additional visualizations of text and image embedding features. Features within the same layer share similar distributions, indicating limited correlation with model inputs or denoising steps. Please zoom in to view finer details.

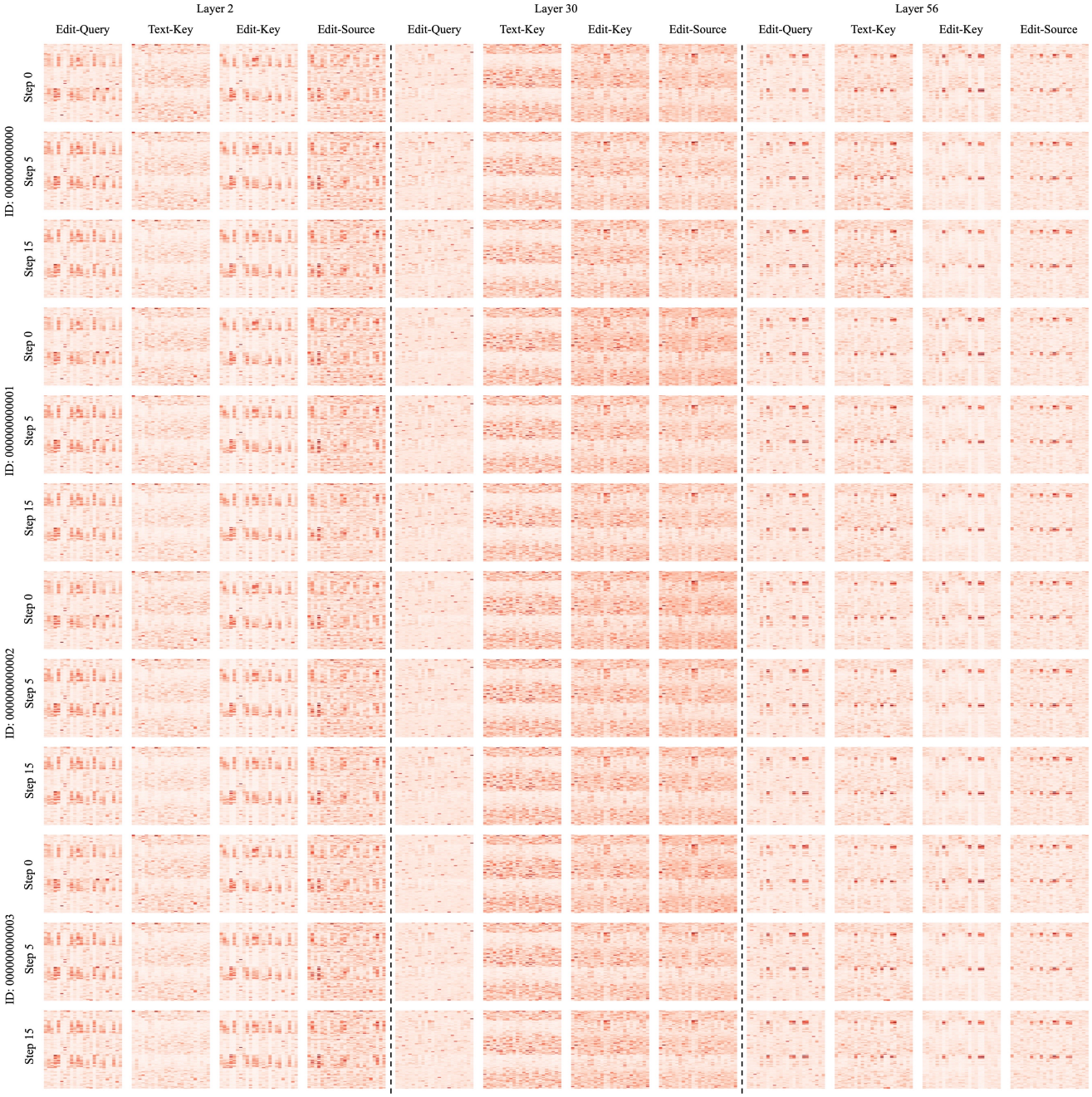


Figure S10. Additional visualizations of aggregating different tokens along the sequence dimension. Please zoom in to view finer details.

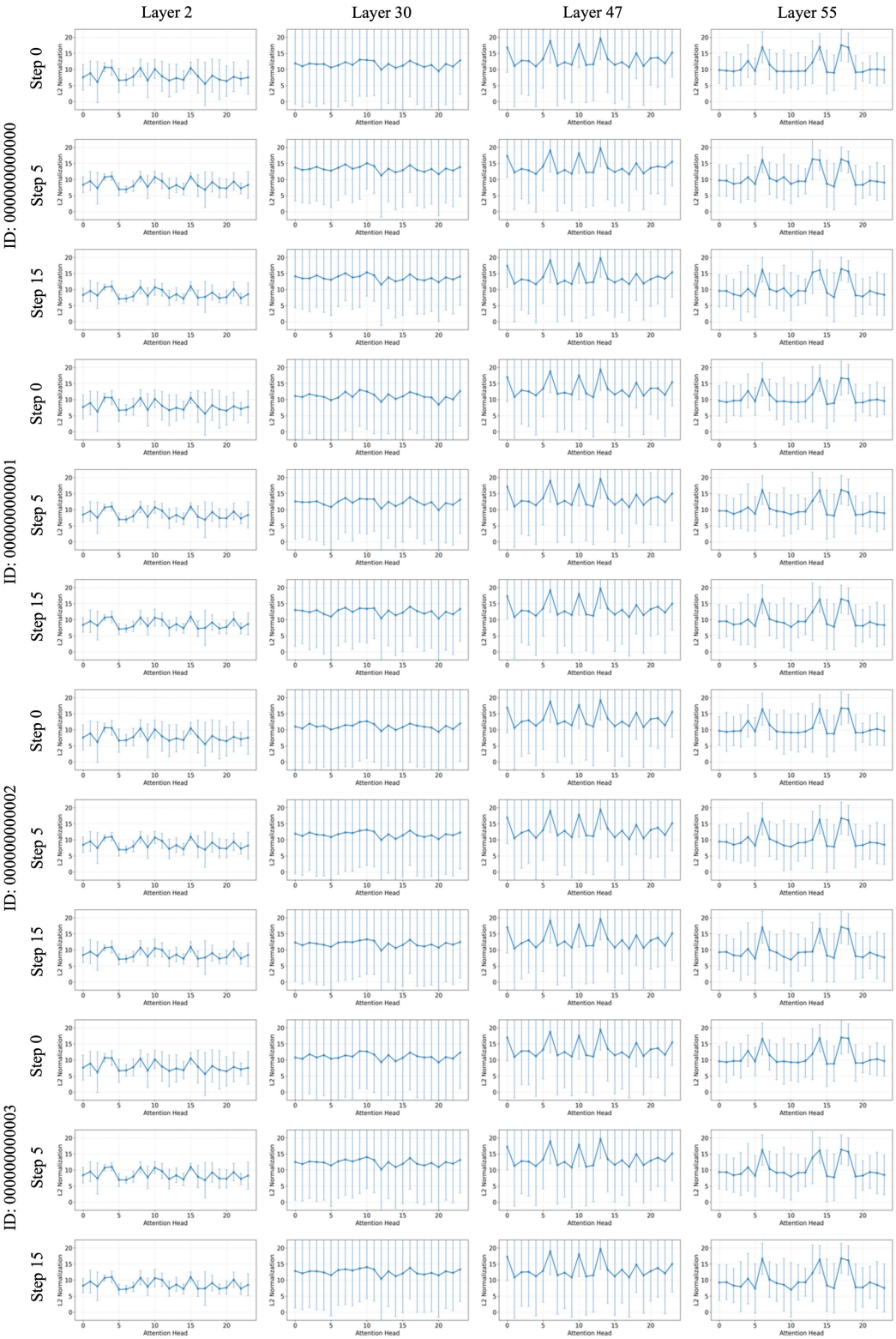


Figure S11. Additional visualizations of Query-edit embedding mean vector magnitudes and standard deviations across different attention heads. Please zoom in to view finer details.

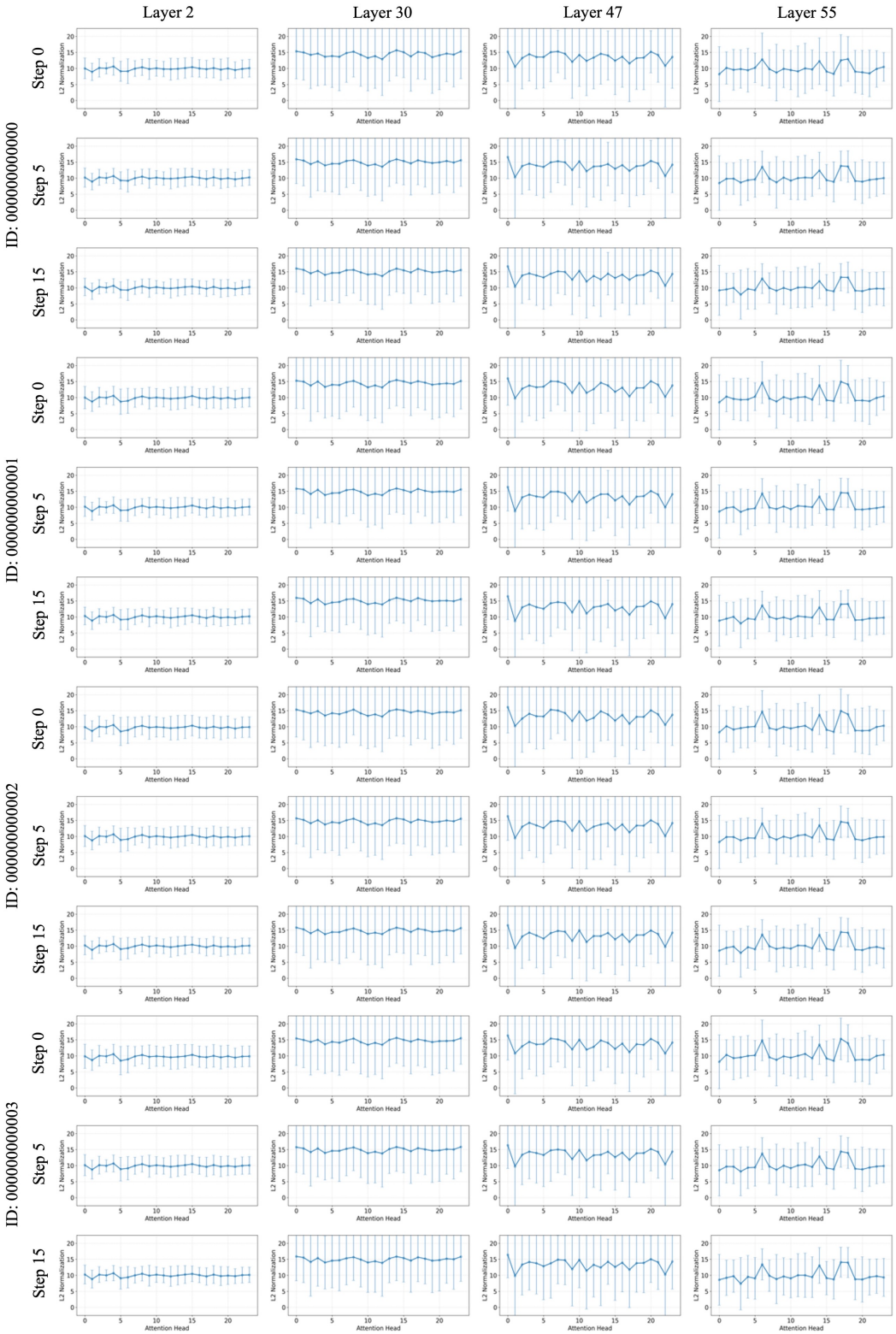


Figure S12. Additional visualizations of Key-text embedding mean vector magnitudes and standard deviations across different attention heads. Please zoom in to view finer details.

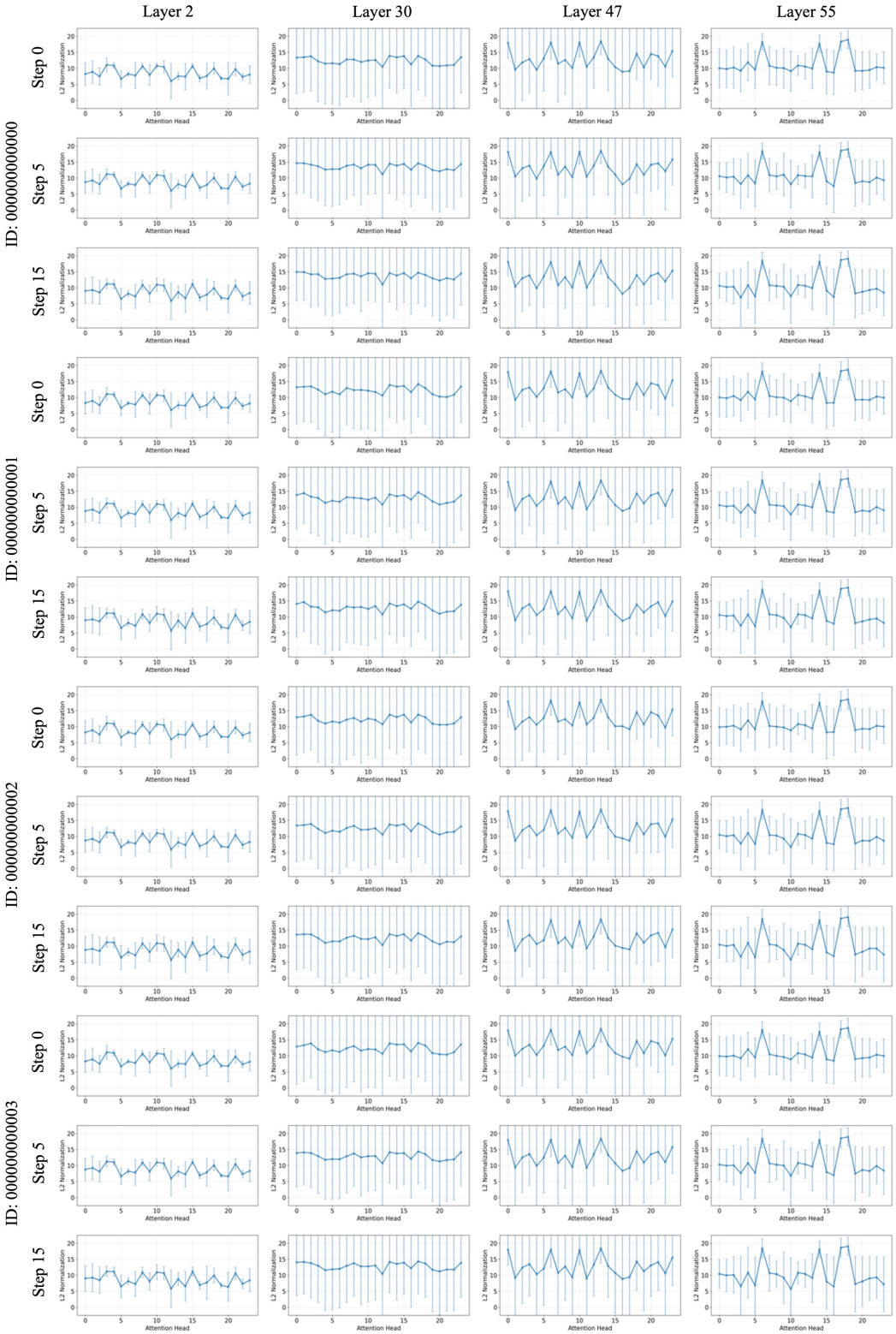


Figure S13. Additional visualizations of Key-edit embedding mean vector magnitudes and standard deviations across different attention heads. Please zoom in to view finer details.

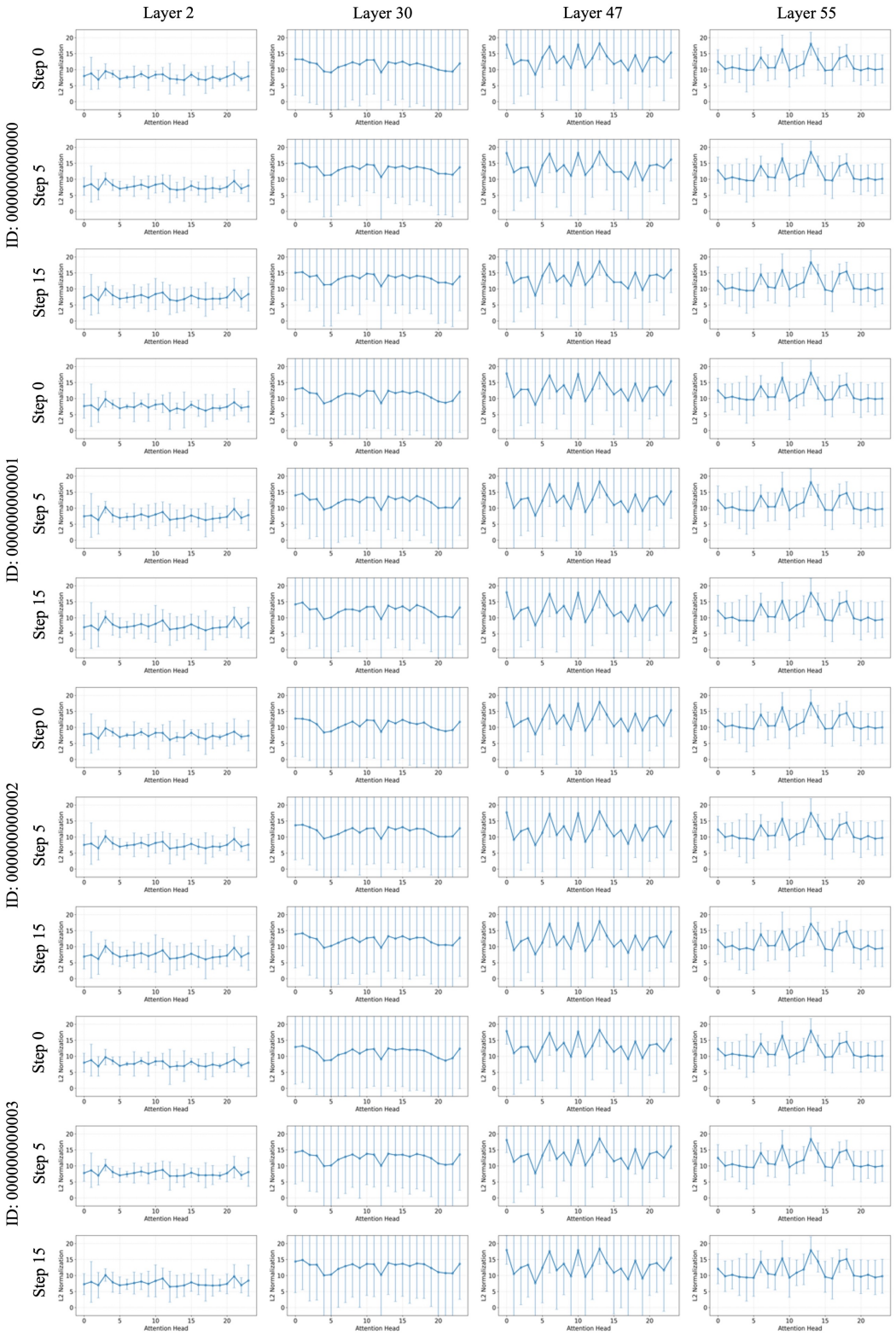


Figure S14. Additional visualizations of Key-src embedding mean vector magnitudes and standard deviations across different attention heads. Please zoom in to view finer details.

G.2. Qwen-Edit Embedding Visualization

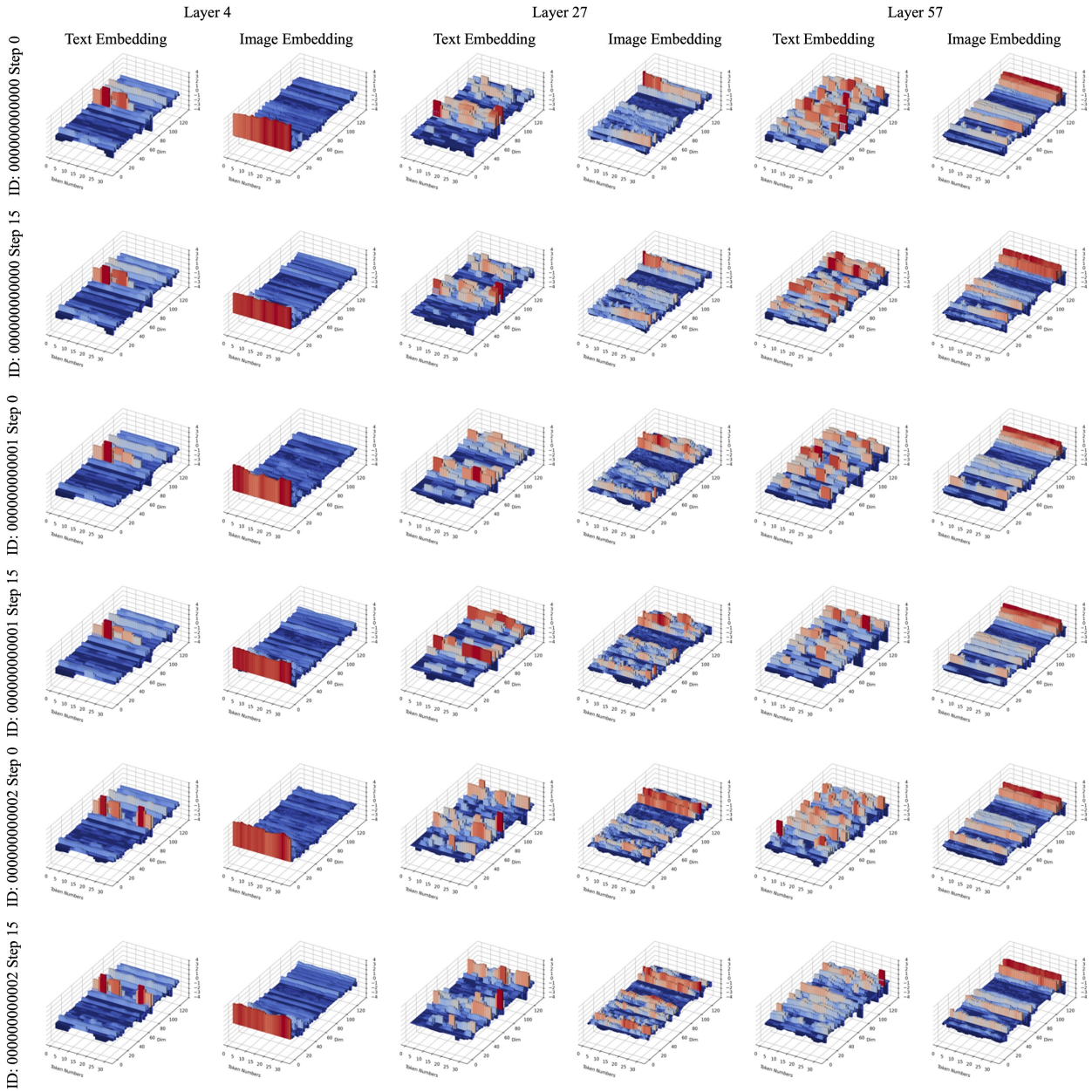


Figure S15. Additional visualizations of text and image embedding features. Features within the same layer share similar distributions, indicating limited correlation with model inputs or denoising steps. Please zoom in to view finer details.

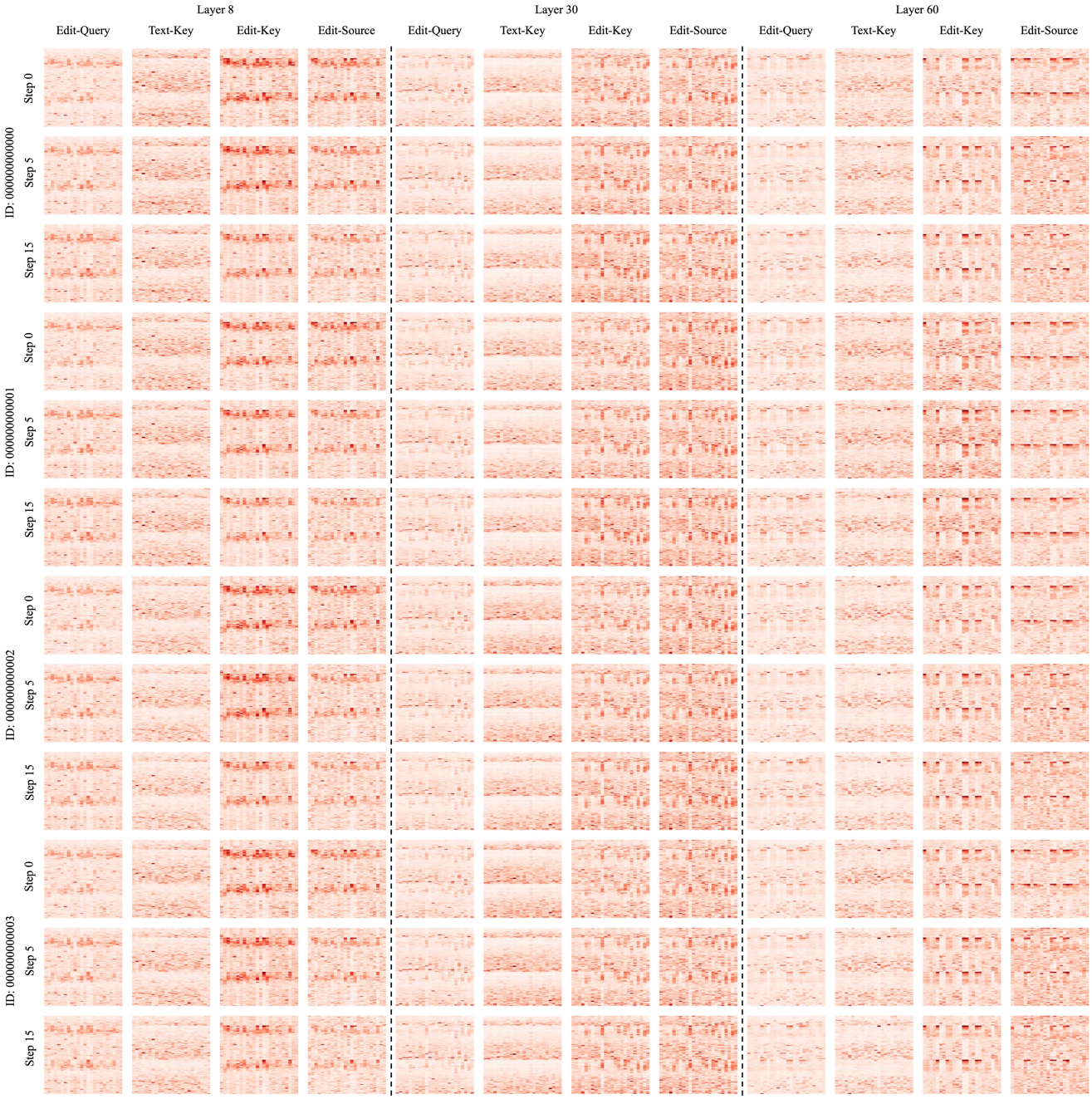


Figure S16. Additional visualizations of aggregating different tokens along the sequence dimension. Please zoom in to view finer details.

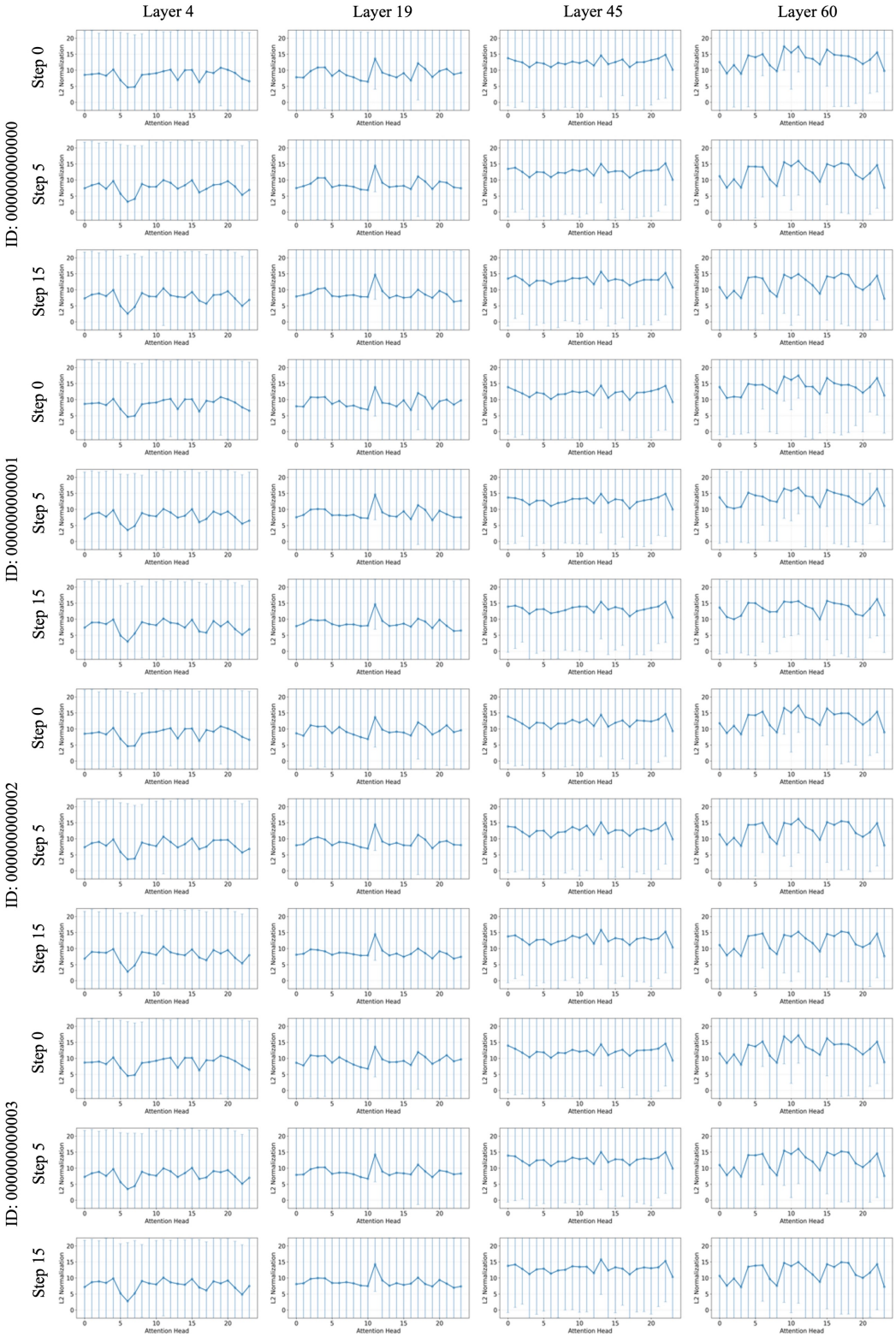


Figure S17. Additional visualizations of Query-edit embedding mean vector magnitudes and standard deviations across different attention heads. Please zoom in to view finer details.

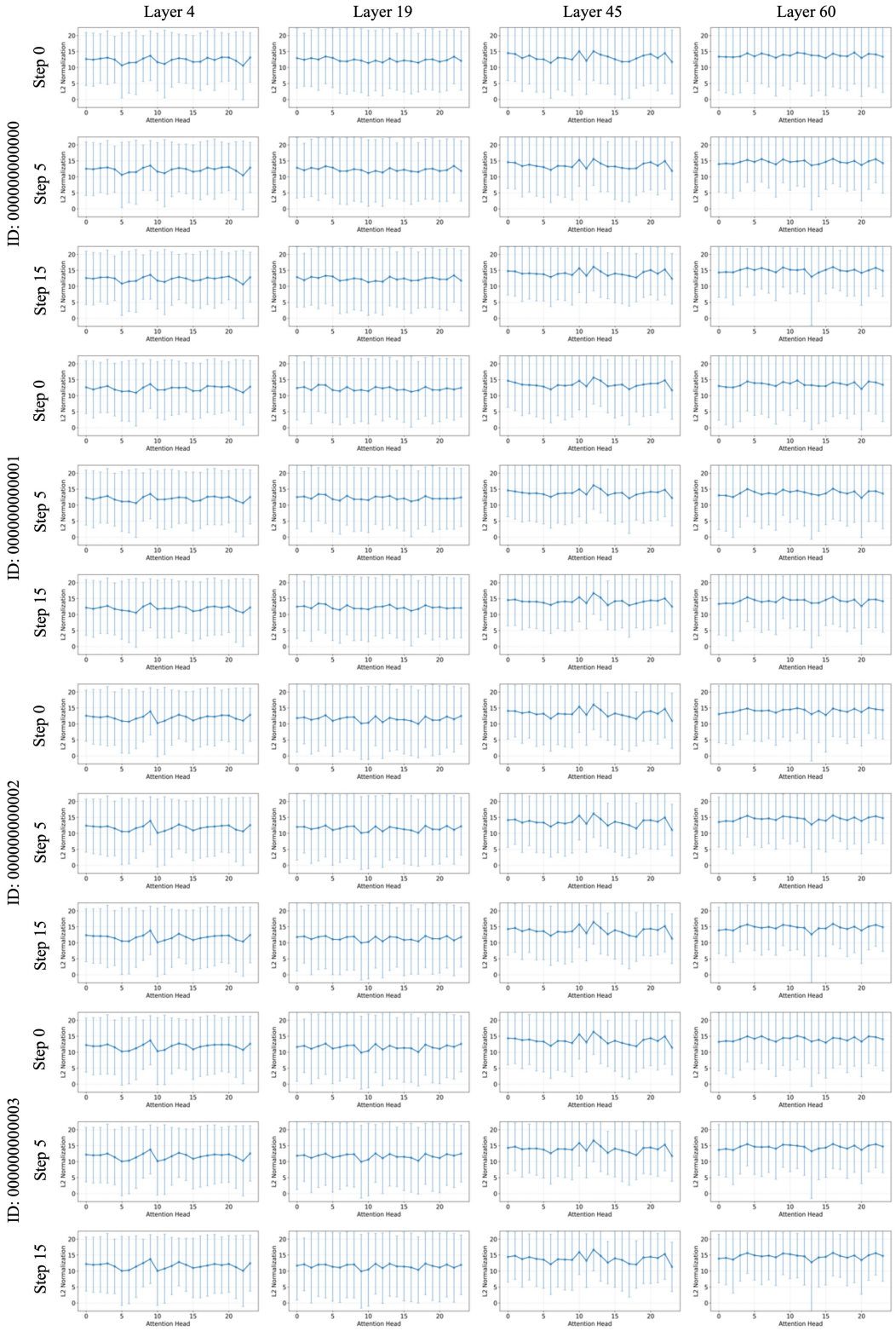


Figure S18. Additional visualizations of Key-text embedding mean vector magnitudes and standard deviations across different attention heads. Please zoom in to view finer details.

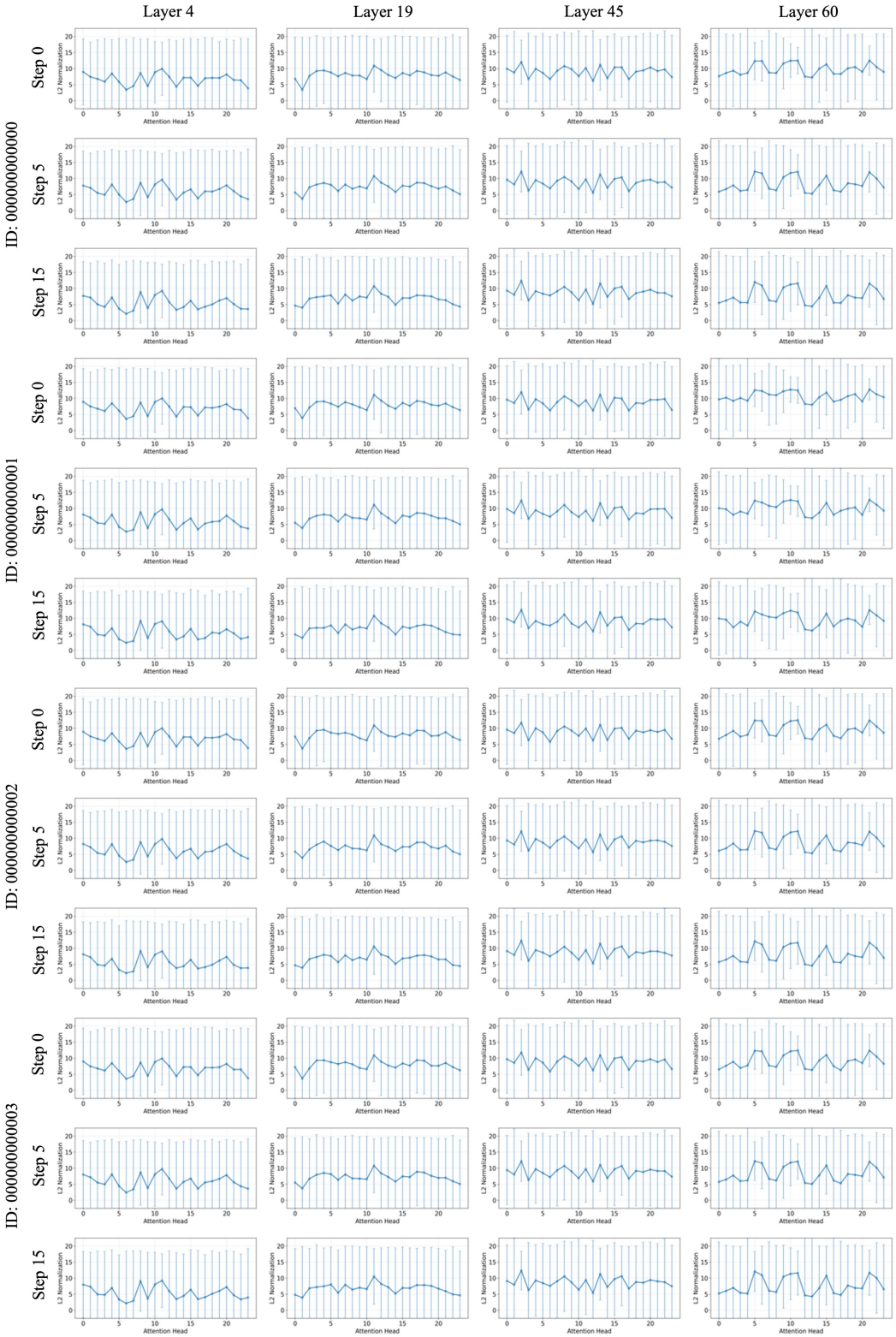


Figure S19. Additional visualizations of Key-edit embedding mean vector magnitudes and standard deviations across different attention heads. Please zoom in to view finer details.

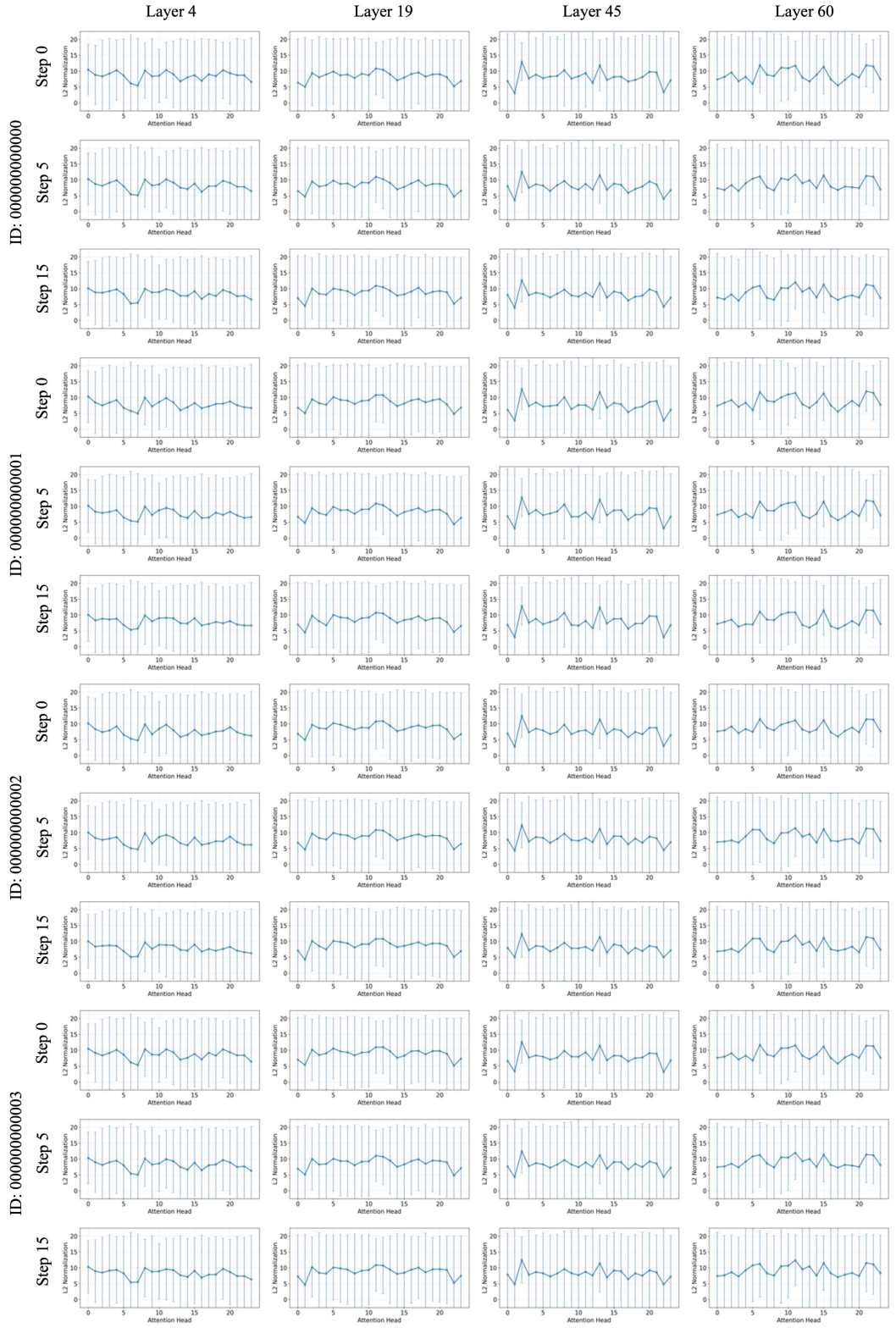


Figure S20. Additional visualizations of Key-src embedding mean vector magnitudes and standard deviations across different attention heads. Please zoom in to view finer details.