UniEdit-I: Training-free Image Editing for Unified VLM via Iterative Understanding, Editing and Verifying

Supplementary Material

1. Additional Settings and Results

1.1. CLIP Score Setting

For the experiments on CLIP score and CLIP feature cosine similarity, we use CLIP-ViT-B/32 as the base model.

1.2. GEdit-Bench Categorize Results

GEdit-Bench. We adopt **GEdit-Bench**, a highly authentic and representative benchmark, as the core framework for systematically evaluating our proposed image editing model. Developed by the Step1X-Edit research team, GEdit-Bench has emerged as a widely recognized and authoritative benchmark in the field of text-guided image editing. It comprises 606 meticulously curated test samples, constructed through a multi-stage pipeline. Over 1000 real-world image editing requests were initially collected from platforms such as Reddit and subsequently filtered to remove redundancy and ensure relevance. Each sample includes an original image, a natural language editing instruction, and a detailed description of the desired outcome, providing rich contextual information for comprehensive evaluation. To enable fine-grained analysis, all samples were manually annotated and categorized into 11 distinct editing task types—such as object manipulation, attribute modification, and style transfer—ensuring balanced task coverage and facilitating rigorous assessment of model performance across diverse editing objectives.

Table 1. GEdit Categorize Results

Task Type	**GEdit-Bench-EN (Full set)** ↑		
	G_SC	**G_PQ**	**G_O**
background_change	8.050	7.650	7.747
color_alter	7.750	7.675	7.462
material_alter	6.275	7.300	6.472
motion_change	7.600	7.550	7.444
ps_human	7.271	7.543	7.290
style_change	7.483	7.467	7.386
subject-add	7.900	7.750	7.668
subject-remove	6.807	7.333	6.739
subject-replace	7.683	7.267	7.335
text_change	4.000	6.434	4.495
tone_transfer	7.900	7.425	7.570
Average	7.156	7.399	7.055

GEdit-Bench Categorize Results. Table 1 reports the subjective evaluation results of GEdit on GEdit-Bench-EN (Full set) across different editing task categories. The metrics include G_SC (semantic consistency), G_PQ (perceptual quality), and G_O (overall performance), all scored on a 0–10 scale, where higher is better. Overall, most task types achieve scores in the range of 6.5–7.8, indicating that our method produces stable and high-quality editing results across diverse scenarios. Tasks such as background_change (G_SC=8.050, G_O=7.747), subject-add (G_SC=7.900, G_O=7.668), and tone_transfer (G_SC=7.900, G_SC=7.570) rank the highest, suggesting strong generalization and visual coherence for background replacement, subject addition, and style transfer edits. In contrast, the text_change task shows significantly lower scores (G_SC=4.000, G_O=4.495), highlighting that text-specific edits remain challenging. this can be attributed to the limitations inherited from the underlying unified vision-language model (VLM), Since our method operates within the pre-trained representation space, it cannot fully overcome these shortcomings during the editing process. As a result, when tasked with modifying or generating precise textual content, the model struggles to produce semantically accurate or visually consistent changes, leading to the observed lower performance on the text_change task.

2. Effectiveness of Dynamic Gain Mechanism

Fixed gain ($\alpha_t = 1.0$) applies a constant editing strength throughout the entire process, lacking awareness of the current semantic alignment progress or whether the editing goal has been achieved. As a result, it is highly prone to *over-editing* or *under-editing*. Specifically, when the target has already been reached, the system continues modifying the image, introducing unnecessary perturbations to irrelevant regions and degrading structural integrity and visual consistency. Conversely, in complex tasks, fixed gain may lead to insufficient editing due to premature termination, failing to fully satisfy the user instruction. This method follows a pre-defined, static trajectory—an open-loop control mechanism—akin to "blindly pushing forward" without leveraging real-time feedback for dynamic correction or adaptive intensity modulation, making precise and reliable editing difficult to achieve.

In contrast, the dynamic gain mechanism fundamentally addresses this limitation by tightly coupling editing intensity with real-time semantic feedback, thereby resolving the misalignment caused by the absence of process perception.

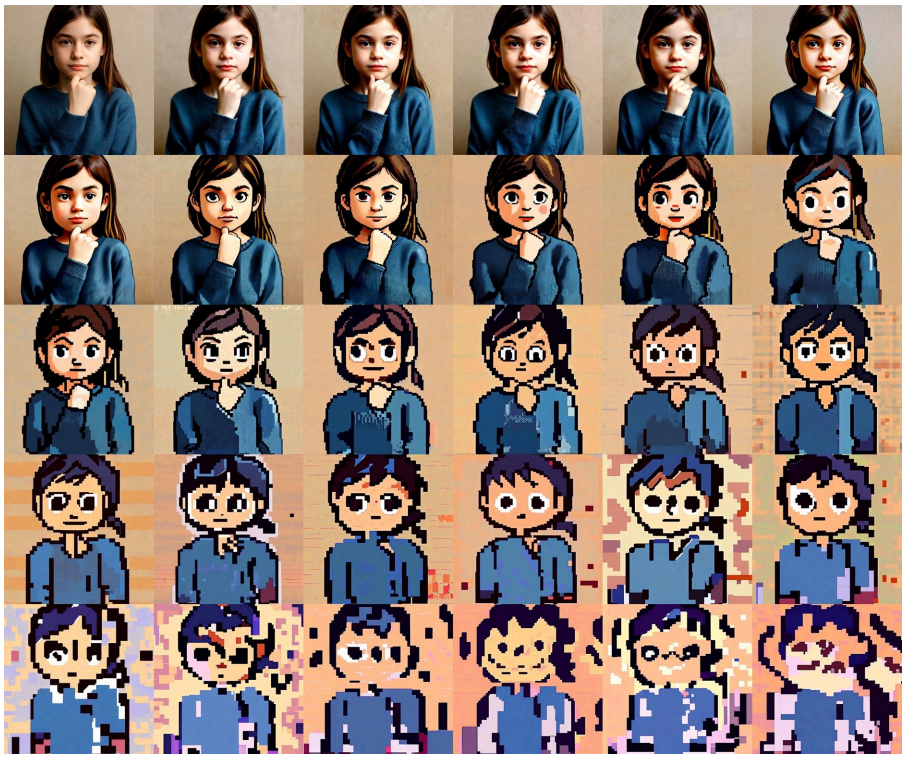


Figure 1. Example of Fixed Gain.

Figure 2. Example of Fixed Gain.

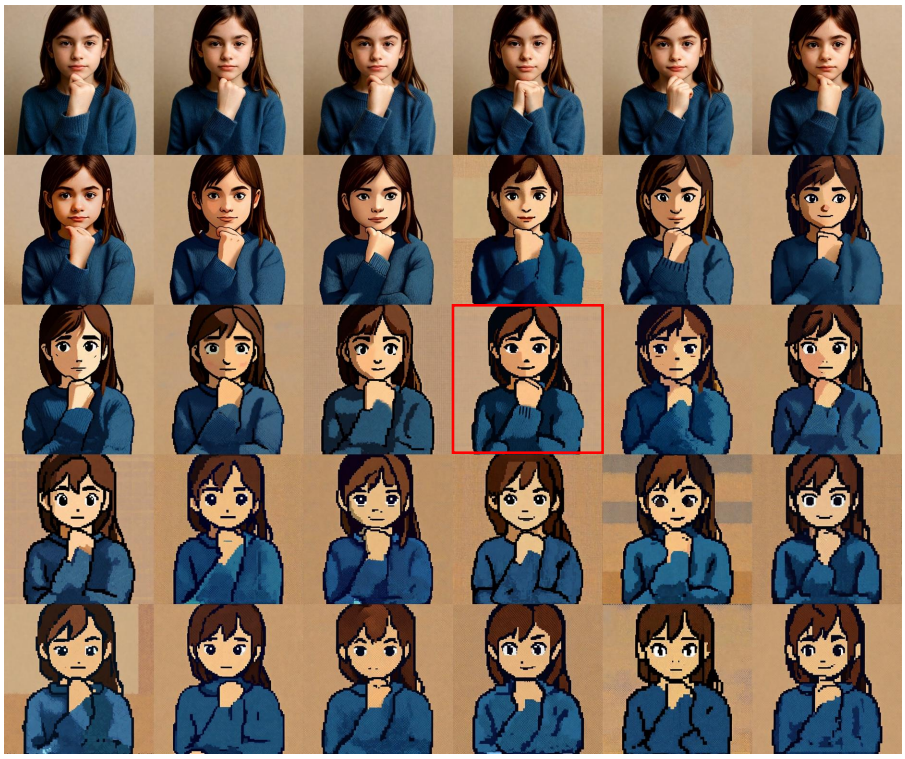


Figure 3. Example of Dynamic Gain.

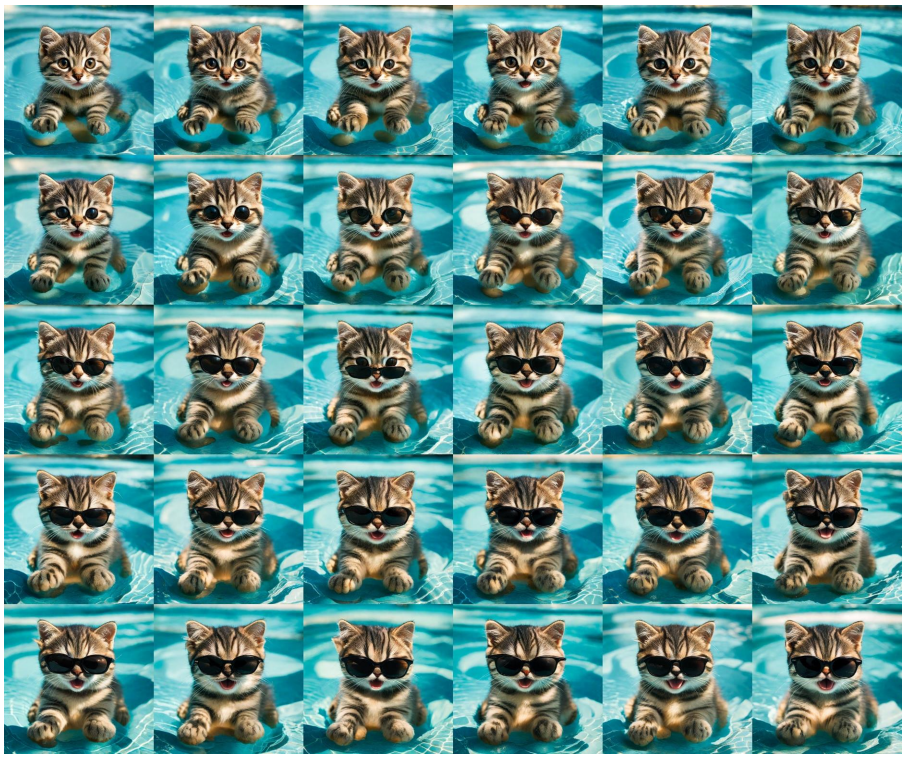


Figure 4. Example of Dynamic Gain.

The gain coefficient is defined as:

$$\alpha_t = \alpha_{\text{base}} \cdot \sigma(\kappa_1 \Delta s_t) \cdot (1 - p_t),$$

where $\alpha_{\text{base}} = 1.0$, $\kappa_1 = 15$, $\Delta s_t = s_t - s_{\text{prev}}$ measures the improvement in global semantic alignment since the last feedback point, $p_t \in [0, 1]$ denotes the task completion score, and $\sigma(\cdot)$ is the sigmoid function. This formulation enables adaptive control: when semantic alignment is improving ($\Delta s_t > 0$), the gain is amplified to accelerate convergence; as the output approaches the target ($p_t \to 1$), the gain is gradually attenuated to prevent over-modification. Consequently, the dynamic gain significantly enhances both the stability of the editing process and the fidelity of the final output.

More importantly, this mechanism works synergistically with UniEdit-I's verification module to form a closed-loop control system. By incorporating multimodal semantic feedback from the VLM every $k = 5$ diffusion steps, and combining it with an early-stopping criterion—halting the process immediately when $s_t > 0.85$ and $p_t > 0.9$ are simultaneously satisfied for two consecutive evaluations—the framework achieves intelligent "stop-upon-success" regulation. This enables automatic adaptation to varying task complexity: simple edits converge rapidly, while complex ones undergo progressive refinement. Crucially, no manual parameter tuning is required. As a result, UniEdit-I realizes efficient, robust, and user-friendly training-free image editing, transforming the process from rigid, open-loop execution into a responsive, self-correcting loop guided by semantic intelligence.

3. More Examples

3.1. From Pixels to Semantics: Editing at the Conceptual Level

As illustrated in Figure 5, 6, and 7, we conduct image editing at the semantic level rather than operating directly in pixel space. This paradigm shift enables a more structured, interpretable, and semantically consistent transformation process. Unlike traditional pixel-level editing—where modifications are applied by directly altering pixel intensities (Figure 5(a), 6(a), and 7(a))—semantic editing operates on high-level representations that encode meaningful attributes such as object identity, pose, texture, and spatial relationships. By manipulating these abstract semantic features, the model adjusts the conceptual definition of the image content, ensuring that each intermediate and final output remains both visually plausible and semantically coherent (Figure 5(b), 6(b), and 7(b)).

One of the key advantages of semantic editing lies in its the visibility and interpretability of intermediate editing steps while performing complex transformations. Pixel-level methods often suffer from artifacts such as blurring, misalignment, or structural distortions when handling edits, especially when multiple changes are required simultaneously. These approaches typically lack an understanding of scene semantics, treating images as mere grids of pixels without awareness of higher-order concepts like object hierarchy or contextual dependencies. In contrast, because semantic editing is grounded in a conceptual understanding of the image content, it can naturally interpret logical combinations of editing commands.

Furthermore, semantic editing supports progressive refinement, enabling smooth transitions from coarse structural adjustments to fine-grained details. This hierarchical editability mirrors human perception and cognitive processing, making the editing process not only more effective but also more intuitive and user-friendly. It also facilitates interpretability, as each editing step corresponds to a meaningful change in semantic space, allowing users to understand and trace the transformation path.

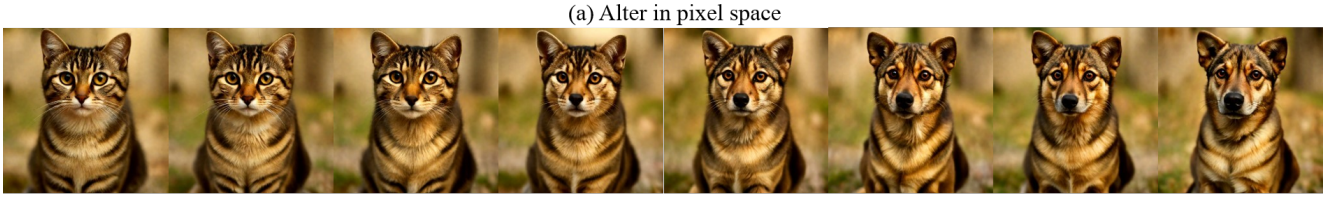
While pixel-level methods remain useful for local retouching or noise-level adjustments, semantic editing offering greater robustness, consistency, and alignment with user intent.

3.2. More Editing Eesults

In conclusion, UniEdit-I establishes a new paradigm for image editing by treating unified vision-language models (VLMs) not as static evaluators, but as active, in-process conductors of semantic refinement. Through a training-free, closed-loop mechanism, our method dynamically steers editing trajectories in the CLIP semantic space, using real-time feedback to iteratively align visual outputs with textual instructions. By exploiting the intrinsic cross-modal alignment of VLMs, UniEdit-I achieves state-of-the-art semantic fidelity without any fine-tuning, demonstrating that robust, intention-aligned editing stems not from model scale or massive curated datasets, but from structured, self-guided reasoning within the latent space. This framework naturally supports a broad spectrum of edits—from coarse structural changes to fine-grained attribute control—and enables complex, multi-instruction compositions within a single coherent process. Unlike black-box end-to-end generators, our approach offers transparent, step-by-step interpretability, clearly revealing how semantic concepts are progressively realized across the editing loop.

We present additional visual results in the supplementary materials, further showcasing UniEdit-I's versatility across diverse tasks. These examples highlight its consistent ability to produce high-fidelity, artifact-free outputs while faithfully adhering to nuanced or compound language prompts, all without task-specific adaptation or retraining. The results reinforce that semantic editing via closed-loop VLM guidance is both powerful and generalizable.

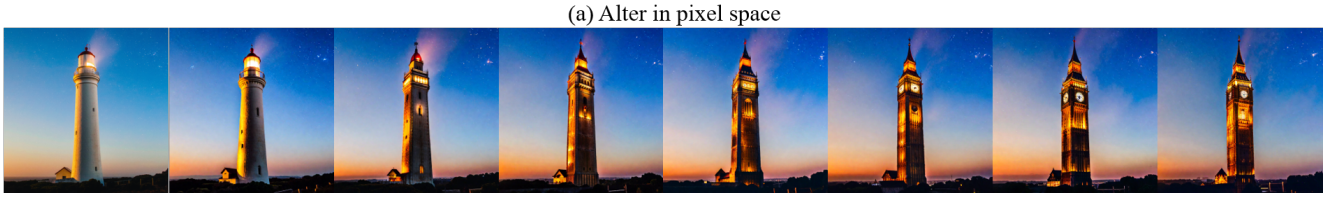
(a) Alter in pixel space



(b) Alter in semantic space

Figure 5. Comparison of (a) Editing in pixel space and (b) Editing in semantic space.

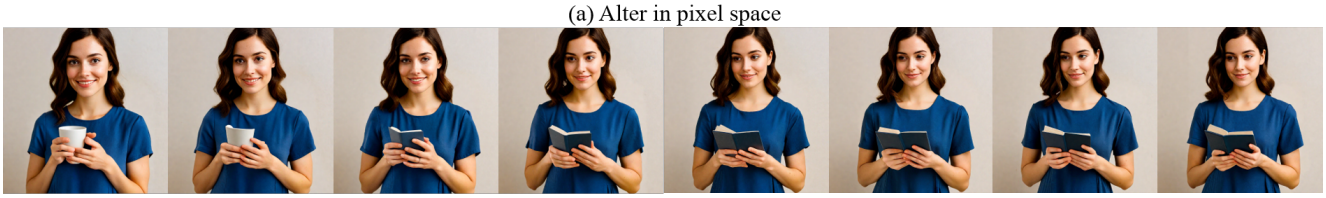
(a) Alter in pixel space



(b) Alter in semantic space

Figure 6. Comparison of (a) Editing in pixel space and (b) Editing in semantic space.

(a) Alter in pixel space



(b) Alter in semantic space

Figure 7. Comparison of (a) Editing in pixel space and (b) Editing in semantic space.

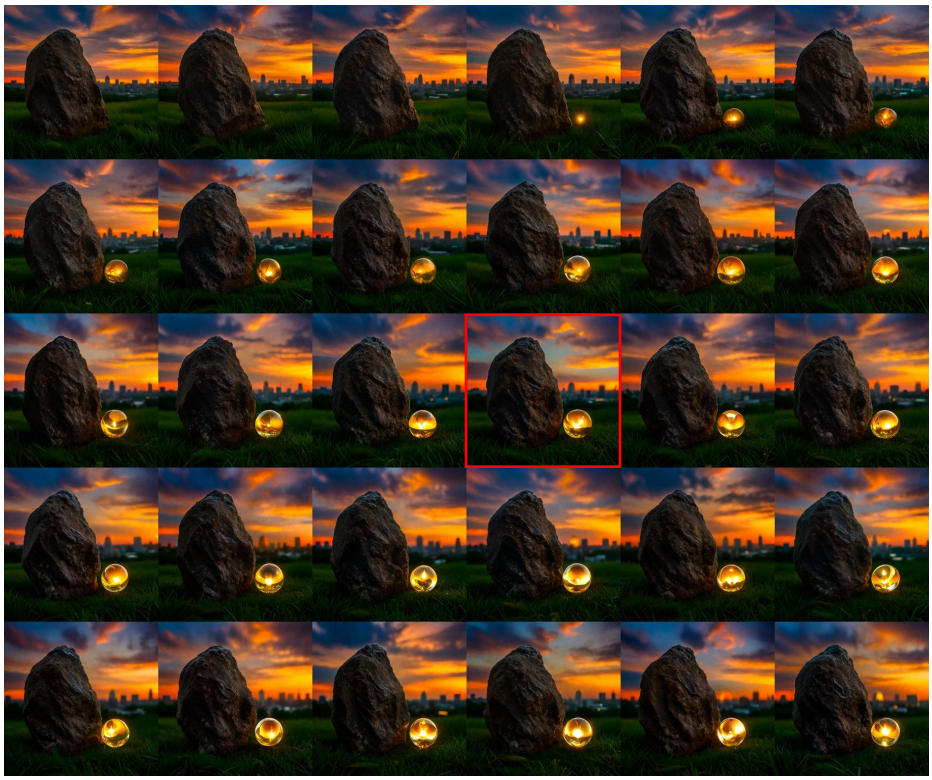


Figure 8. Subject Add Task.

Figure 9. Subject Change Task.

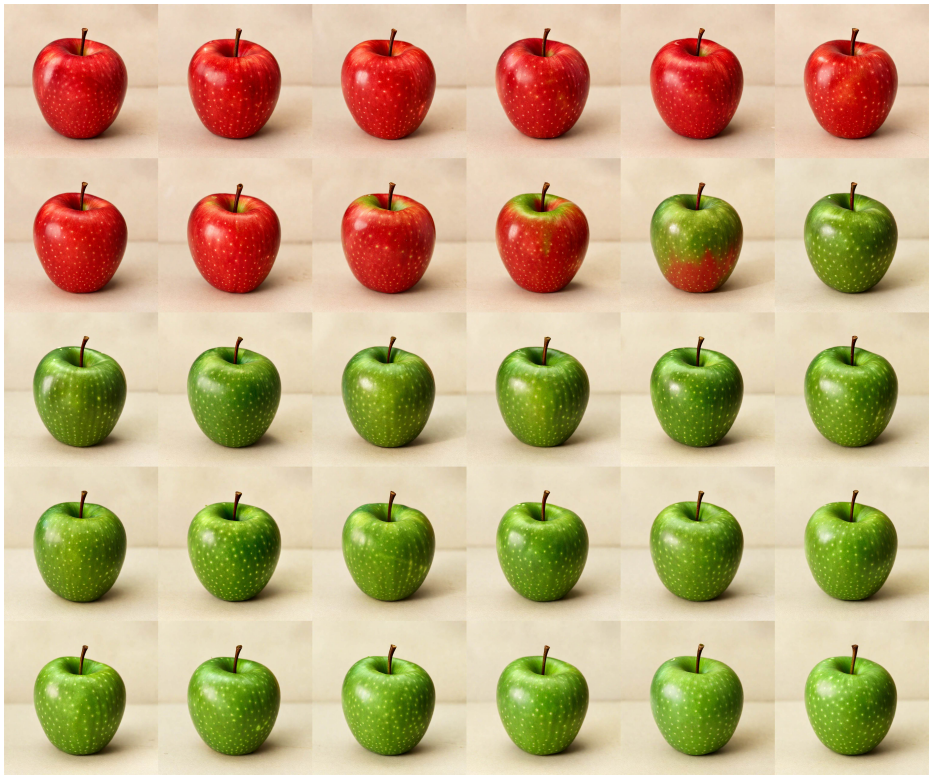


Figure 10. Color Alter Task.

Figure 11. Background Change Task.

Figure 12. Attribute Change Task.

Figure 13. Motion Change Task.

Figure 14. Style Change Task.

Figure 15. Multi-Task Editing.