UnicEdit-10M: A Dataset and Benchmark Breaking the Scale-Quality Barrier via Unified Verification for Reasoning-Enriched Edits

Supplementary Material

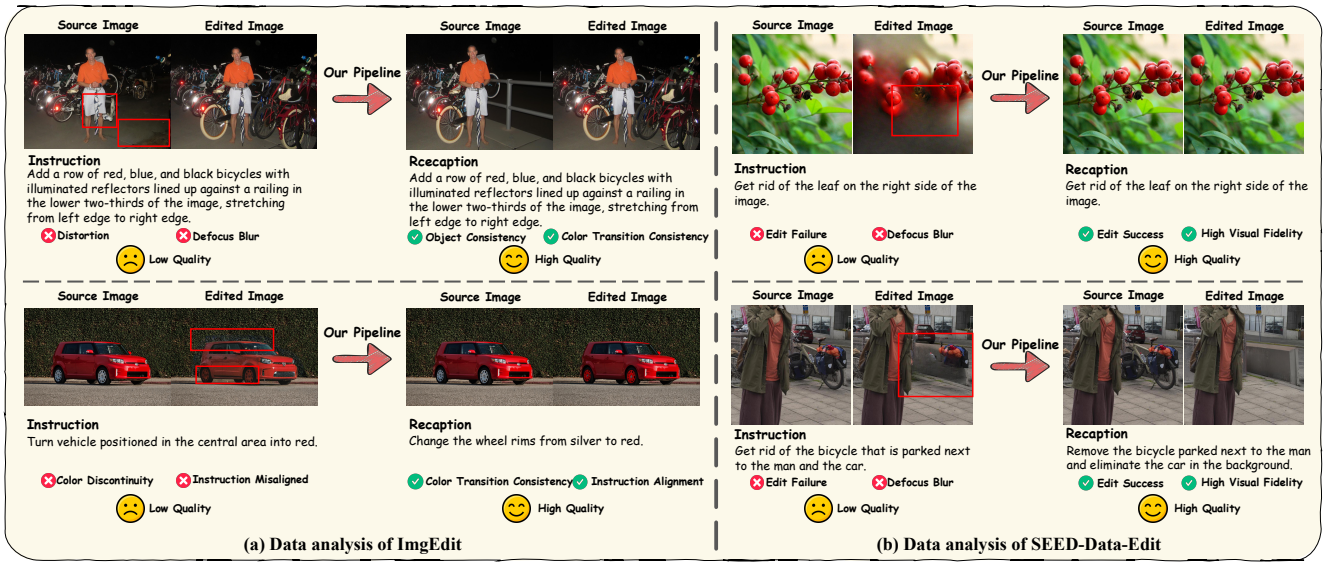

Figure 5. Qualitative comparison of data curation pipelines. (a) shows example triplets from ImgEdit [46]. (b) shows example from SEED-Data-Edit [12]. In each subfigure, the left columns display original triplets, while the right columns show the same images reprocessed through *Our Pipeline*. Red boxes highlight regions of blurring, artifacts, and color inconsistencies in the originals. Our pipeline consistently yields higher quality and more precisely aligned instructions, demonstrating the effectiveness of our unified verification process.

In this supplementary material, we further elaborate on three core components of our framework: the data construction pipeline, the expert verification model, and the UnicBench benchmark. For the **Data Construction Pipeline**, we present the dataset taxonomy in Sec. A, analyze our pipeline and post-verification procedures in Sec. B, discuss noisy cases in Sec. E, and study facial consistency in Sec. F. For the **Expert Model Evaluation**, we describe the experimental setup and metrics used to assess our verification model in Sec. C. For the **Benchmark Analysis**, we validate the evaluation metrics in Sec. D, report comprehensive benchmark results in Sec. G, compare UnicBench with existing benchmarks in Sec. H, and provide qualitative examples and evaluation prompts in Secs. J and K.

A. Dataset Taxonomy

To establish our taxonomy, we surveyed prominent datasets, including ImgEdit [46], Step1X-Edit [23], and SEED-Data-Edit [12], to consolidate and reclassify their editing tasks. We further introduced a novel reasoning category to enhance complex reasoning capabilities and address gaps in existing benchmarks. This process yielded a comprehensive classification of 22 unique editing tasks, spanning from basic object manipulation to complex semantic reasoning, as detailed in Tab. 6.

B. Data Pipeline Analysis

B.1. Comparison with Multi-toolchain Pipelines

Automated data curation pipelines [12, 23, 46, 50] that rely on a sequence of vision tools are susceptible to error propagation, where inaccuracies from upstream modules are amplified in downstream processes, ultimately degrading dataset quality. To illustrate this, Fig. 5 presents a qualitative analysis of samples from ImgEdit [46] and SEED-Data-Edit [12], both of which employ such multi-toolchain architectures. The examples exhibit common artifacts stemming from this approach, including significant blurring and inconsistent color blending at edit boundaries. Furthermore, Fig. 5(a) highlights a critical issue of misalignment between the instruction and the resulting edit.

We processed these challenging samples through our proposed pipeline to demonstrate its corrective capabilities. The results show a marked improvement in image quality, with artifacts eliminated and semantic coherence re-

Table 6. Definitions of editing taxonomy of UnicEdit-10M.

Task	Subtask	Description
Object Editing	Subject Addition	Adding a new object or person to the image.
	Subject Removal	Removing a specified object or person from the image.
	Subject Replacement	Replacing an object with another.
	Object Extraction	Isolating and extracting a target object from its background.
	Counting Change	Modifying the number of objects in the image.
	Text Modification	Editing textual content within the image.
	Portrait Editing	Enhancing or modifying facial features.
Attribute Editing	Color Alteration	Changing the color of an object or region.
	Material Modification	Changing the material properties of an object.
	Motion Change	Adjusting the dynamic pose or action of an object.
	Texture Editing	Modifying the surface texture details of an object.
	Shape-Size Alteration	Changing the shape or size of an object.
Scene Editing	Background Change	Replacing or modifying the image background.
	Style Transfer	Converting the overall image to a target artistic style.
	Tone Transformation	Adjusting the image's color tone and atmosphere.
	Viewpoint Transformation	Changing the camera's viewpoint or position.
	Lens Zooming	Simulating an optical lens zoom effect.
Reasoning Editing	Spatial Reasoning	Moving objects based on spatial logic.
	Multi-Object Coordination	Coordinately modifying the attributes or positions of multiple objects.
	Compound Operations	Executing multiple, combined editing operations.
	Relation Change	Modifying the interactive relationship between objects.
	Implicit Change Edits	Inferring the actual editing task based on context and real-world knowledge.

stored. Concurrently, our expert model automatically re-captioned the instructions to align precisely with the actual visual transformations, thereby correcting the instruction-image mismatch and validating the efficacy of our unified verification and refinement process.

B.2. Comparison of No Edit Filtration Methods

A critical step in our post-verification pipeline is the accurate filtration of *No Edit* samples, where the output image is visually identical or nearly identical to the original. Traditional pixel-based metrics like the Structural Similarity Index (SSIM) [39] are ill-suited for this task, as they lack semantic understanding and are sensitive to imperceptible artifacts introduced by generative models.

Fig. 6 illustrates this deficiency. SSIM [39] proves insensitive to semantically meaningful but visually subtle changes. For instance, the addition of a small bottle (top-left example) yields a high SSIM [39] of 0.9474, incorrectly suggesting no change occurred. Conversely, SSIM [39] is overly sensitive to minor, imperceptible pixel shifts inherent to the editing model's process. In the top-right and bottom-left examples, no visible edit was made, yet their SSIM [39] scores are lower than the sample with a clear object addition, leading to false positives. An even more extreme case (bottom-right example) shows a minor color artifact causing the SSIM [39] to decline to 0.3241.

In contrast, our expert model demonstrates robust, semantically-aware judgment. It correctly identifies the subtle yet valid edit in the first example while accurately classifying the other visually unchanged pairs as *No Edit*, ignoring trivial pixel-level noise. This capability is crucial for reliable, large-scale data curation. The bar chart in Fig. 6 provides quantitative validation, confirming that our expert model achieves superior performance in identifying *No Edit* instances compared to baseline methods.

B.3. Comparison with Other Datasets

We provide a comprehensive comparison with existing datasets in Tab. 1 of the main text. Unlike other datasets, UnicEdit-10M employs a unified post-verification stage that not only filters low-quality samples but also refines instructions to ensure precise alignment with the actual edits. Furthermore, UnicEdit-10M provides extensive data for complex edits, offering broader functional diversity and greater scale than currently available resources.

C. Expert Model Evaluation

To evaluate Qwen-Verify, we engaged two human experts to curate a total of 390 samples, comprising 300 *Normal* cases, 50 *No Edit* cases, and 40 *Hallucination* cases. For all samples, the experts manually revised any erroneous descrip-

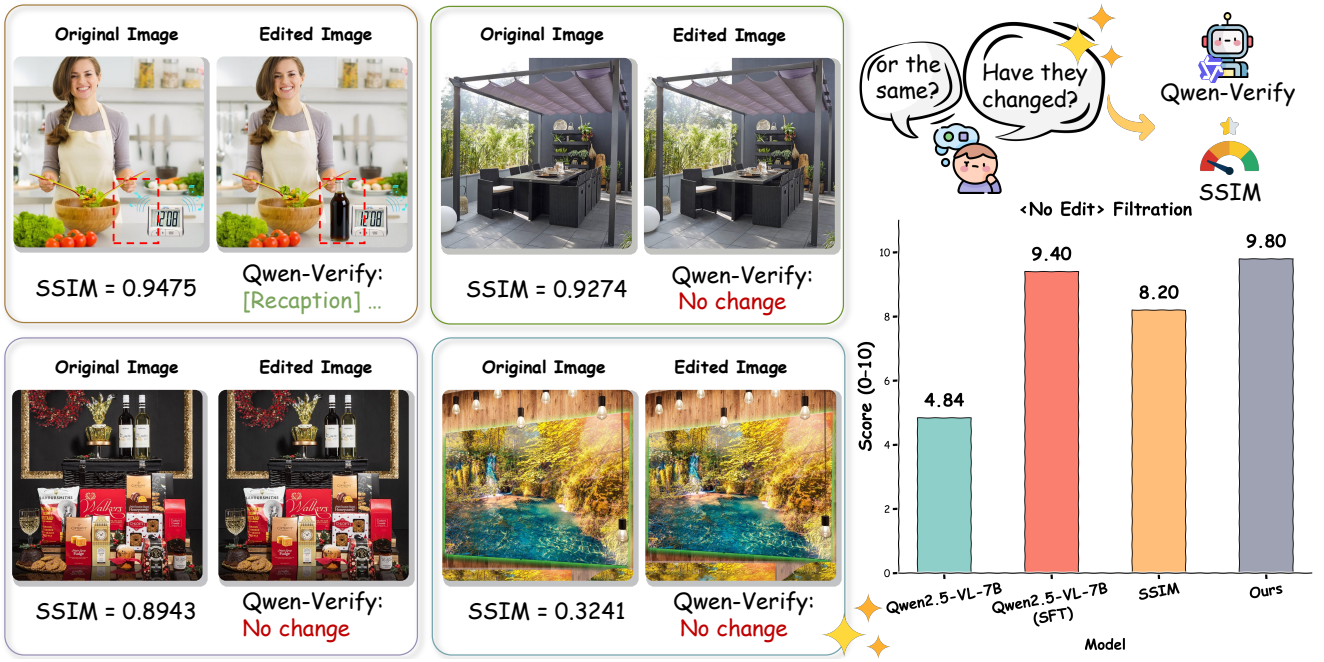

Figure 6. Examples of *No Edit* filtration. The left side shows four examples with detection results from SSIM [39] and Qwen-Verify. The top-left example has a clear edit (red dashed box) but receives a high SSIM [39] score, while the other three visually unchanged pairs receive lower scores. The right side provides a quantitative comparison, confirming that Qwen-Verify performs best at identifying failed edits.

tions to establish reliable ground-truth instructions. To systematically evaluate the performance of Qwen-Verify, we designed an automated evaluation protocol using GPT-4.1. This protocol parses atomic editing tasks from each instruction and computes a score to objectively quantify the alignment between the ground-truth instruction and the instruction generated by our model.

Atomic Task Decomposition. We formalize an instruction P as a set of atomic editing tasks, T_P . Each atomic task t_i is a tuple representing a core operation:

$$P \xrightarrow{\text{parse}} T_P = \{t_i\}_{i=1}^{N}, \text{ where } t_i = (o_i, a_i) \tag{5}$$

In this formulation, o_i denotes the target object or region, and a_i represents the corresponding edit action.

Alignment Accuracy. Given the set of atomic tasks from the ground-truth instruction, T_{GT}, and the set from the generated instruction, T_{GEN}, we compute an *Alignment Accuracy (Acc.)*. The score measures the coverage of ground-truth tasks while penalizing for hallucinated or redundant tasks. It is defined as:

$$\text{Acc}(T_{GEN}, T_{GT}) = \underbrace{\frac{|T_{GT} \cap T_{GEN}|}{|T_{GT}|}}_{\text{Coverage (Recall)}} - w \cdot \underbrace{\frac{|T_{GEN} \setminus T_{GT}|}{|T_{GT}|}}_{\text{Redundancy Penalty}} \tag{6}$$

Here, the first term calculates the recall of correctly identified atomic tasks. The second term imposes a penalty, weighted by w, for any extraneous tasks generated by the model. In our experiments, we set $w = 0.5$. This metric provides a quantitative measure of the precision and fidelity of the generated answers. After validation, the expert model is deployed in our pipeline for large-scale data curation.

D. Validation of Benchmark Metrics

Our evaluation protocol offers a fine-grained assessment by decomposing edit quality into four key dimensions: *Instruction Following (IF)*, *Non-edit Consistency (NC)*, *Visual Quality (VQ)*, and *Reasoning Accuracy (RA)*. This approach offers a more diagnostic alternative to other metrics like the VIEScore [19], which can obscure specific failure modes.

Comparison with VIEScore. As illustrated in Fig. 8, our protocol demonstrates superior sensitivity to common editing failures. In example (a), the model replaces the man instead of adding a woman to his left. While VIEScore [19] assigns an SC score of 9, failing to penalize this critical error in non-edit preservation, our NC metric correctly identifies the unintended removal and assigns a lower score of 7. Similarly, in example (b), changing 'CLASSIC MOJITO' to 'BABY MILKSHAKE' also results in collateral damage to the text below it. VIEScore [19] again overlooks

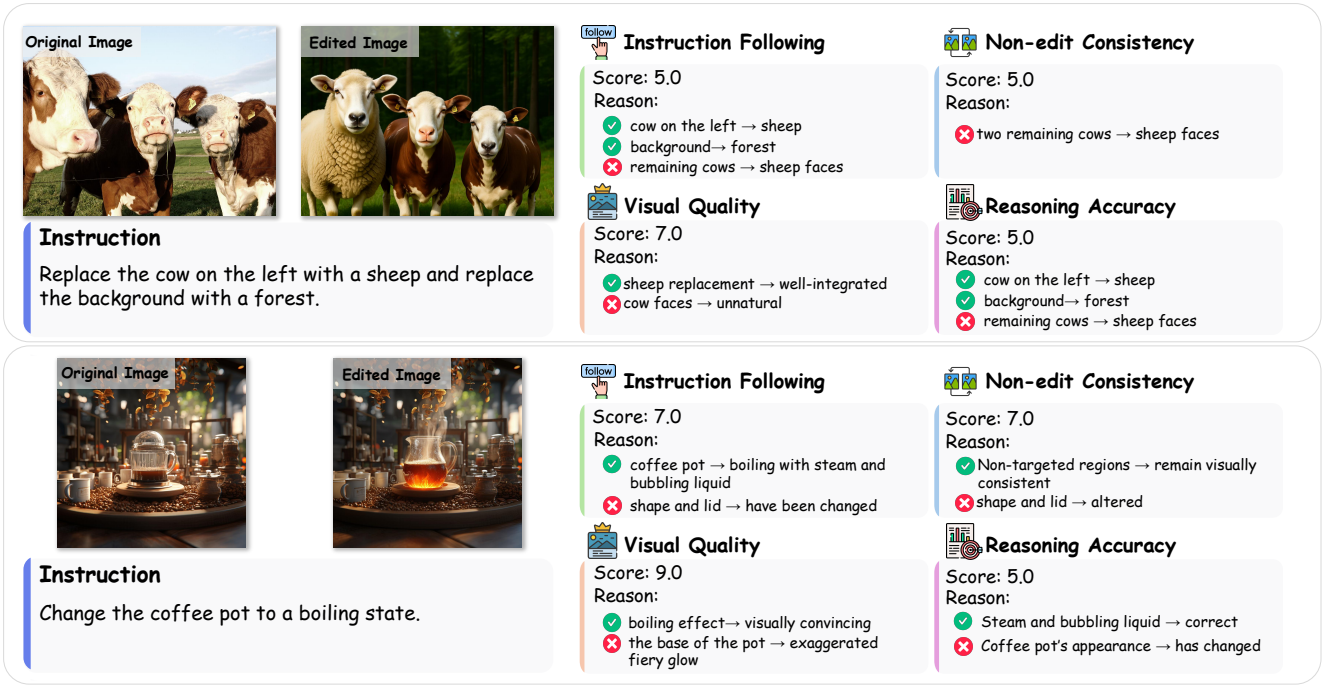

Figure 7. Examples of our evaluation protocol on two complex edit cases. For each example, the edited image pair and instruction are shown on the left. The right side displays the scores for four evaluation dimensions, along with the detailed reasons from the VLM evaluator, which specifies points of credit and deduction.

Table 7. Comparison of different image editing benchmarks.

Benchmarks	#Size	#Subtasks	Human Filtering	Basic Edit	Complex Edit	Public Available
EditBench [37]	240	1	✘	✔	✘	✔
EmuEdit [32]	3055	7	✘	✔	✘	✔
HIVE [49]	1000	1	✔	✔	✘	✔
HQ-Edit [16]	1640	7	✘	✔	✘	✔
MagicBrush [48]	1053	7	✔	✔	✘	✔
AnyEdit-Test [47]	1250	25	✘	✔	✘	✔
ICE-Bench [26]	6538	31	✔	✔	✘	✘
ImgEdit-Bench [46]	737	9	✔	✔	✔	✔
GEdit-Bench [23]	606	11	✔	✔	✘	✔
KRIS-Bench [43]	1267	22	✔	✘	✔	✔
UnicBench	1100	22	✔	✔	✔	✔

this whereas our NC metric penalizes the collateral change. These cases show that our fine-grained metrics more reliably assess preservation of non target regions.

Alignment with Human Judgment. To validate that our VLM-based evaluation aligns with human judgment, we analyze its scoring on complex cases in Fig. 7. In the top example, the instruction involves two tasks: changing the leftmost cow to a sheep and the background to a forest. While the model executes these tasks, it incorrectly alters two other cows into sheep. Our protocol accurately diagnoses this partial failure: the IF score is lowered to 5.0 because the instruction was not precisely followed (only the *leftmost* cow was specified), and the NC score is also 5.0, penalizing the unintended modifications to the other animals. The VQ score is 7.0, reflecting good overall image quality but noting the unnatural facial features of the edited animals. Finally, guided by the *reasoning-points list*, the RA metric correctly identifies the erroneous object-type change, resulting in a score of 5.0.

The bottom example tests the model's reasoning capability by requesting to make the coffee pot "boil". The model correctly adds steam and bubbles but undesirably alters the pot's shape and adds an exaggerated flame. Both IF and

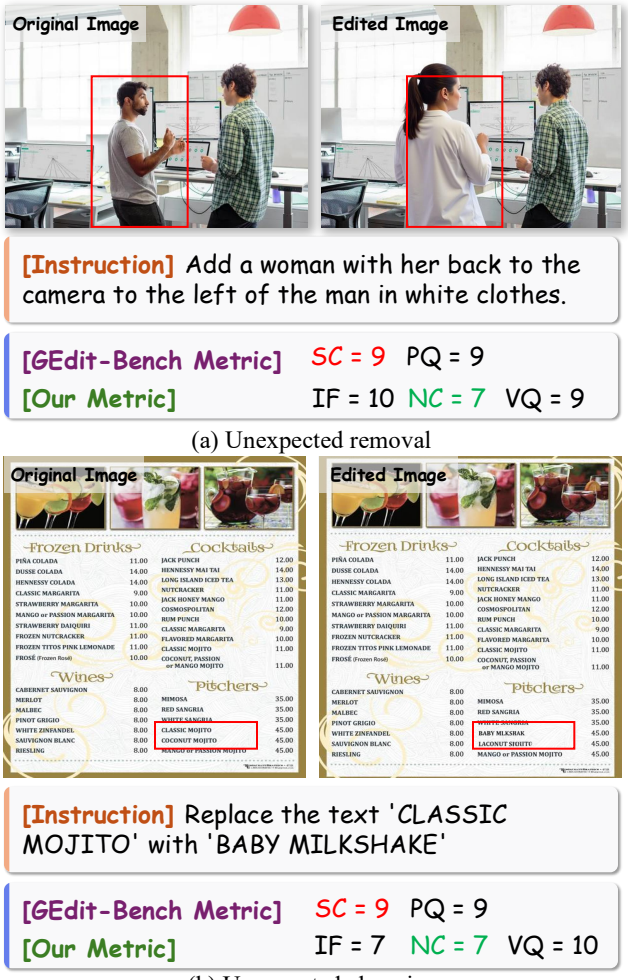

Figure 8. Comparison between GEdit-Bench's [23] metrics and UnicBench's metrics. (a) compares scoring for a case with an unexpected removal. (b) compares scoring for a case with an unexpected text change.

NC scores are set to 7.0, acknowledging the partial success while penalizing the unintended shape distortion. The VQ score is reduced due to the unrealistic flame. The RA score is 5.0, recognizing that the concept of "boiling" was successfully rendered but penalizing the failure to preserve the object's core identity (its shape). These case studies demonstrate that our multi-dimensional metric, as judged by a VLM, provides a comprehensive and human-aligned assessment, effectively pinpointing the specific strengths and weaknesses of an editing model in a single evaluation.

E. Analysis of Noisy Data

As outlined in Sec. 3.2, our initial data generation process produces noisy triplets that fall into three primary categories: (1) *Edit Failure*, where no discernible edit is made; (2) *Instruction-Image Misalignment*, where the visual change deviates from the instruction; and (3) *Others*, which encompasses a range of quality issues.

Our expert model, Qwen-Verify, is specifically trained to address the first two, more prevalent issues by filtering null edits and recaptioning misaligned ones, thereby ensuring semantic and instructional fidelity. For the third category, which includes issues like low aesthetic quality, anatomical deformities (*e.g.*, incorrect limbs on human subjects), and other structural distortions, we employ pre-trained aesthetic scoring models and specialized human-body detectors to automatically flag and remove such low-quality samples. This multi-stage refinement strategy ensures that the final dataset is not only semantically aligned but also meets a high standard of visual quality.

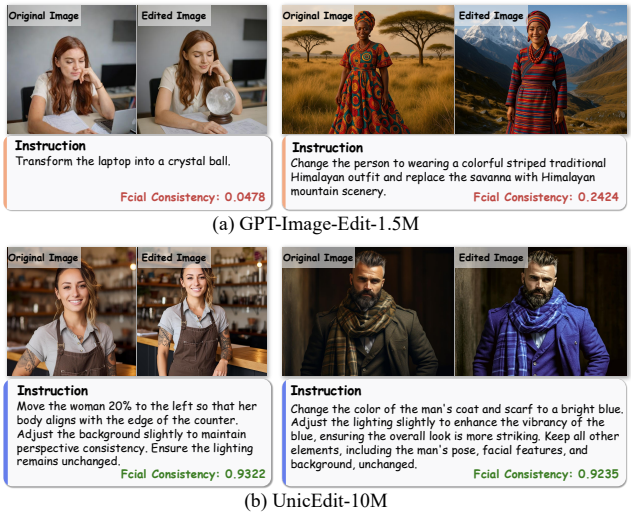

Figure 9. Qualitative comparison of facial consistency. (a) shows examples from GPT-Image-Edit-1.5M [38]. (b) shows examples from UnicEdit-10M.

F. Analysis of Facial Consistency

As reported in the main text, our dataset and GPT-Image-Edit-1.5M [38] achieve comparable Semantic Consistency scores. This prompted a deeper investigation into the qualitative differences between the two datasets. A sampling analysis of GPT-Image-Edit-1.5M [38] revealed a frequent occurrence of facial identity inconsistency, a critical issue

Table 8. Comparison of facial consistency between GPT-Image-Edit-1.5M [38] and UnicEdit-10M.

Datasets	Facial Consistency
GPT-Image-Edit-1.5M [38]	0.3025
Ours	**0.8911**

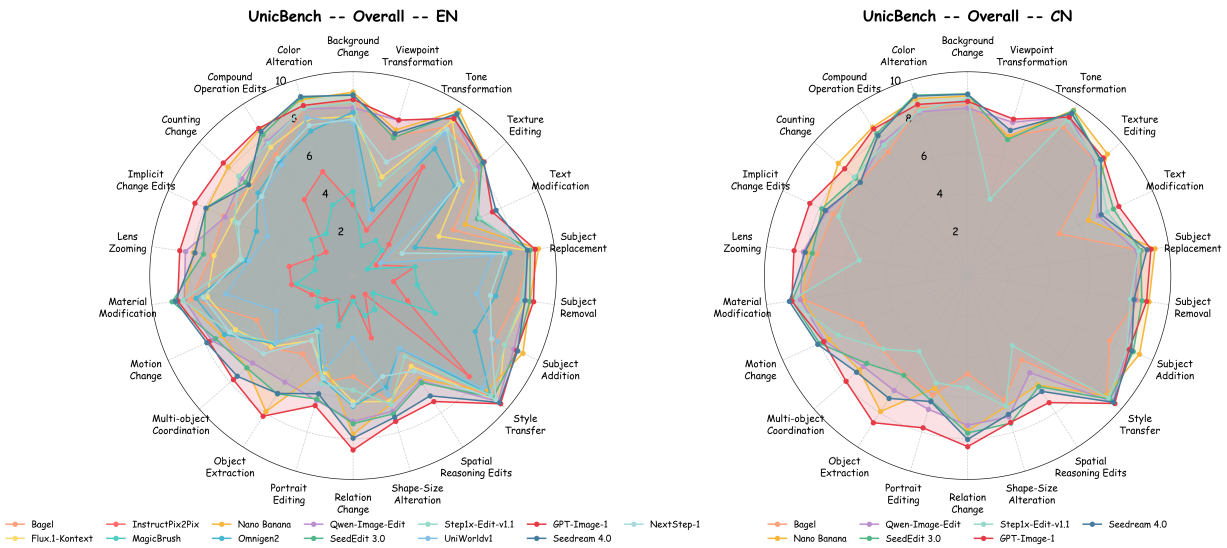

Figure 10. Overall score of each model on the sub-tasks in UnicBench, for EN (left) and CN (right) instructions. All results are evaluated by *GPT-4.1*.

Table 9. Detailed performance across different editing tasks (EN). The performance of open-source and closed-source models is separately marked with the best performance in **bold**, and the second best underlined. All results are evaluated by *GPT-4.1*.

Model	Attribute Editing				Object Editing				Scene Editing				Reasoning Editing				
	IF	NC	VQ	Overall	IF	NC	VQ	Overall	IF	NC	VQ	Overall	IF	NC	VQ	RA	Overall
Open-Source Models																	
Instruct-Pix2Pix [4]	3.1320	4.3880	4.3840	3.2273	2.0143	3.3152	3.3754	2.1075	4.2880	5.7240	4.8600	4.5773	2.3080	3.2760	3.4840	1.9560	2.0990
MagicBrush [48]	2.3655	3.6466	3.6185	2.4559	2.5229	3.7000	3.6514	2.6074	2.3120	2.7320	3.1960	2.1907	2.0880	3.3360	3.2800	1.7240	2.0027
OmniGen2 [42]	7.1600	7.9080	6.8640	6.9011	5.5514	7.1400	6.3514	5.6477	6.6880	7.9680	6.4560	6.6075	5.8600	7.1160	6.3400	5.1240	5.5329
UniWorld-v1 [22]	5.8080	8.1480	7.0200	6.3074	4.8457	6.6886	6.1486	5.1441	6.1360	7.5480	6.8840	6.4598	4.6160	7.1000	6.0120	4.0160	4.6767
FLUX.1-Kontext [3]	6.8760	**8.9160**	7.4880	7.0269	6.4286	**8.1486**	7.2800	6.5028	7.7040	**8.4960**	7.6200	7.6453	6.2320	**8.4560**	7.0840	5.5040	6.1634
NextStep-1 [34]	7.8600	6.3160	6.9320	6.7591	6.3000	5.5229	6.0029	5.6326	8.4056	7.1165	7.0241	7.3901	6.7360	6.0400	6.2480	6.2920	6.1626
BAGEL [8]	7.5440	8.8120	7.3400	7.3184	6.6343	7.9771	7.0800	6.6784	8.9200	8.2400	7.5960	8.1083	6.1440	7.8520	6.5640	5.2600	5.9326
Step1X-Edit-v1.1 [23]	7.7560	8.7280	7.8440	7.6342	6.7886	8.1400	7.4314	6.9090	7.4880	7.9280	7.2280	7.2437	6.0280	8.0480	6.8120	5.0400	5.8982
Qwen-Image-Edit [41]	**8.3960**	8.2800	**8.0720**	**7.9378**	**7.8400**	7.8029	**8.0657**	**7.5128**	**9.2960**	8.3440	**8.5800**	**8.6277**	**7.4360**	7.7680	**7.5840**	**6.4480**	**6.9168**
Closed-source Models																	
Nano Banana [7]	7.7000	**9.3040**	8.2160	7.9117	7.6319	**9.0348**	8.2174	7.7034	8.7400	**8.8760**	8.4800	8.5360	7.9600	**8.6880**	7.8600	6.8680	7.4324
SeedEdit 3.0 [35]	8.5265	8.9959	8.0082	8.2087	7.7609	8.3382	7.7230	7.6277	9.1215	8.1417	8.2672	8.4011	7.8785	8.2632	7.4049	6.9393	7.3269
Seedream 4.0 [31]	8.6160	9.1040	8.1880	**8.3225**	7.8257	8.5286	7.9200	7.6482	8.9960	8.8560	8.3280	8.5813	8.2880	8.4680	7.9200	7.5960	7.7770
GPT-Image-1 [17]	**9.1532**	7.7957	**8.6000**	8.2475	**8.8448**	8.0090	**8.7373**	**8.2422**	**9.6680**	8.1618	**8.8631**	**8.8009**	**9.0705**	7.3172	**8.4978**	**8.3392**	**8.1576**

for practical applications. To quantify this observation, we conducted a comparison of facial consistency. For portrait image pairs from both datasets, we employed RetinaFace [10] for face detection and ArcFace [9] to extract feature vectors, subsequently calculating the cosine similarity between the faces in the source and edited images. The results, presented in Tab. 8, quantitatively confirm that our dataset maintains a significantly higher face consistency score. Qualitative examples in Fig. 9 further illustrate this distinction. In the examples from GPT-Image-Edit-1.5M [38] shown in Fig. 9 (a), the left image pair shows noticeable changes in the woman's facial features and face shape, and the right image pair shows a change in the woman's perceived ethnicity. In contrast, our dataset, as shown in Fig. 9 (b), exhibits much stronger facial identity preservation. This evidence indicates that our data generation pipeline better preserves subject identity throughout the editing process, enhancing the dataset's overall quality and utility.

G. Analysis of Benchmark Results

G.1. Overall Analysis

While Tab. 4 of the main text presents the aggregate scores for mainstream models, we visualize their performance breakdown by sub-task in the radar chart in Fig. 10. The chart clearly reveals a performance dichotomy: most models perform well on foundational tasks like *Background Change* and *Subject Replacement*, but all models exhibit a significant performance drop on *Viewpoint Transformation*, *Text Modification*, and *Spatial Reasoning Edits*. This indicates that current architectures still lack robust spa-

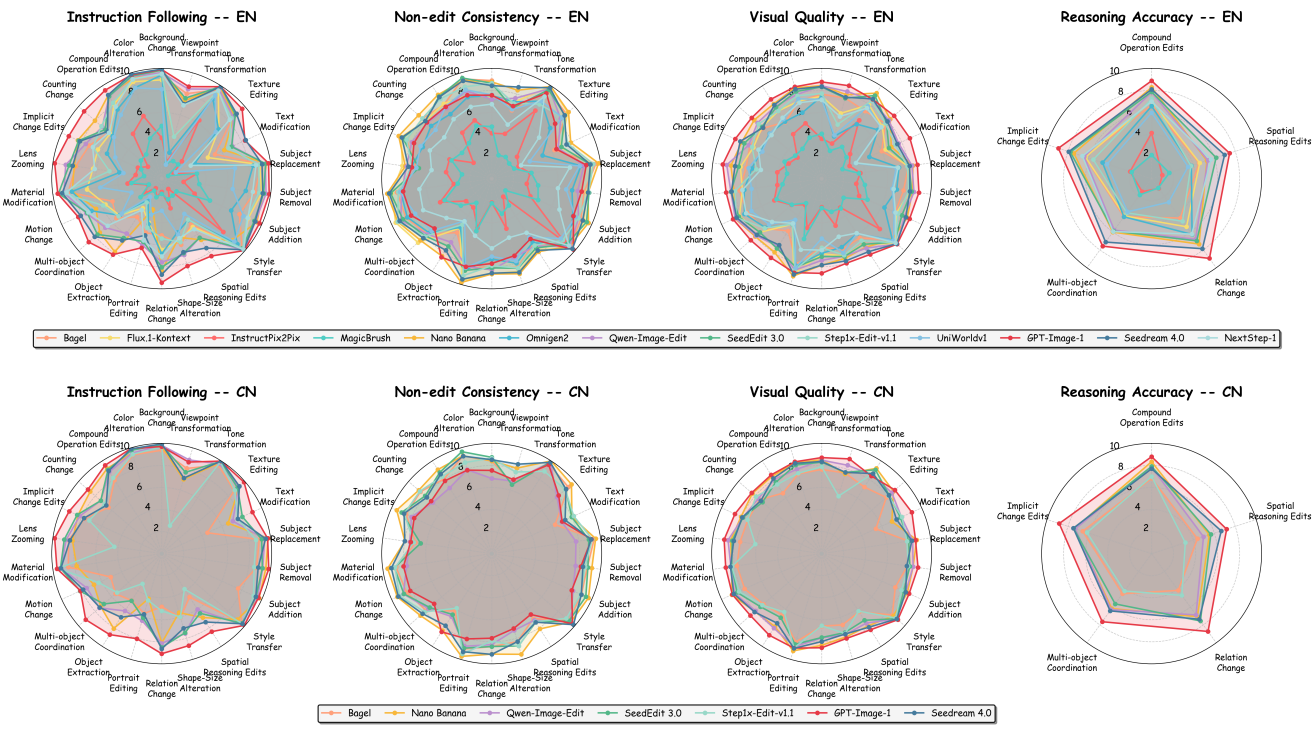

Figure 11. Performance of four evaluation dimensions for each sub-task. The top row shows results for EN tasks, and the bottom row shows results for CN tasks. All results are evaluated by *GPT-4.1*.

Table 10. Detailed performance across different editing tasks (CN). The performance of open-source and closed-source models is separately marked with the best performance in **bold**, and the second best underlined. All results are evaluated by *GPT-4.1*.

Model	Attribute Editing				Object Editing				Scene Editing				Reasoning Editing				
	IF	NC	VQ	Overall	IF	NC	VQ	Overall	IF	NC	VQ	Overall	IF	NC	VQ	RA	Overall
Open-source Models																	
BAGEL [8]	7.5560	8.8120	7.5080	7.4498	6.7114	8.1829	7.3029	6.9385	9.0360	8.2000	7.6880	8.1469	6.1400	7.9840	6.7520	5.2840	5.9540
Step1X-Edit-v1.1 [23]	8.0080	**8.8640**	7.9360	7.8829	6.9229	**8.4171**	7.8429	7.1484	7.2600	**8.2800**	7.4000	7.2448	5.9640	**8.0840**	6.9480	5.0560	5.9374
Qwen-Image-Edit [41]	**8.6200**	7.9400	**8.2240**	**8.0001**	**7.8971**	7.5971	**8.2286**	**7.5682**	**9.4200**	8.1120	**8.5920**	**8.5920**	**7.7400**	7.6320	**7.7960**	**6.6560**	**7.0400**
Closed-source Models																	
Nano Banana [7]	7.8040	**9.4760**	8.4640	8.1122	8.0836	**9.0317**	8.3026	8.0299	8.7800	8.7960	8.4480	8.5037	7.9800	**8.8760**	8.1120	6.8960	7.4994
SeedEdit 3.0 [35]	8.7854	9.0040	8.1457	**8.4379**	7.8006	8.3900	7.8240	7.6794	9.2418	8.2213	8.4590	8.5126	7.8807	8.2016	7.5473	6.8395	7.3807
Seedream 4.0 [31]	8.6840	9.0080	8.2760	8.3881	7.8114	8.3771	7.9657	7.6935	9.0520	**9.9280**	8.3600	8.6731	8.0320	8.4400	8.0120	7.1240	7.5765
GPT-Image-1 [17]	**9.2876**	7.8541	**8.6481**	8.3420	**9.1381**	8.0150	**8.7688**	**8.4681**	**9.6292**	8.2625	**8.8375**	**8.8071**	**9.0925**	7.3524	**8.4978**	**8.2247**	**8.1594**

tial awareness and the ability to precisely edit in-image text. Furthermore, the visualization highlights a critical gap: open-source models lag behind their closed-source counterparts, particularly on complex reasoning edits. This suggests that the primary bottleneck for open-source models is not basic instruction following, but the advanced inference and world knowledge required for complex manipulation.

G.2. Analysis across Different Tasks and Metrics

A detailed analysis of model performance across different editing tasks is provided in Tab. 9 and Tab. 10. For basic tasks, the performance gap between top open-source models like Qwen-Image-Edit [41] and closed-source models is narrow. On some tasks, open-source models even achieve comparable results. For example, Qwen-Image-Edit [41] surpasses Nano Banana [7] in *Attribute Editing* and matches GPT-Image-1 [17] in *Scene Editing*. This suggests that for attribute and scene-level transformations, the capabilities of open-source and closed-source models are converging. However, a substantial performance gap remains in reasoning-intensive tasks, where closed-source models maintain a clear advantage. In this category, Qwen-Image-Edit [41], despite being the SOTA open-source model, lags significantly behind all closed-source models on the IF and RA metrics. This divergence points to the heightened demand for advanced semantic understanding and logical reasoning in complex edits, capabilities that are more densely embedded in closed-source foundation models.

Table 11. Detailed performance across different editing tasks (EN). The performance of open-source and closed-source models is separately marked with the best performance in **bold**, and the second best underlined. All results are evaluated by *Qwen2.5-VL-72B* [1].

Model	Attribute Editing				Object Editing				Scene Editing				Reasoning Editing				
	IF	NC	VQ	Overall	IF	NC	VQ	Overall	IF	NC	VQ	Overall	IF	NC	VQ	RA	Overall
Open-source Models																	
Instruct-Pix2Pix [4]	4.3520	3.8800	6.0560	2.4986	3.5029	3.3114	5.6400	1.9398	5.2880	3.6360	5.9120	3.1577	3.0320	2.8480	5.3400	2.4800	1.3405
MagicBrush [48]	4.2080	2.9480	5.2080	2.6521	4.2229	3.0600	5.8600	2.7003	4.1840	2.4040	5.4400	1.9384	3.1160	2.8200	5.3720	2.4880	1.6255
OmniGen2 [42]	7.3760	6.8440	7.8560	6.3789	6.3543	6.6714	7.4829	5.5498	7.3880	5.7560	7.5720	5.3390	7.2400	7.3080	8.0080	6.2160	5.7325
UniWorld-v1 [22]	5.5800	7.8920	7.1400	5.0760	5.1343	6.4657	7.1057	4.6795	6.5000	**7.2640**	7.0400	5.4614	5.3520	7.5760	7.1760	4.0920	3.8848
FLUX.1-Kontext [3]	6.6400	**8.3120**	7.6840	6.1066	7.0629	**8.2400**	8.1400	6.5634	7.7520	6.7280	7.9320	6.4008	7.3600	**8.4040**	8.2800	6.2600	5.9478
NextStep-1 [34]	8.4240	6.5560	8.1440	7.0760	7.6714	5.7800	7.8886	6.1585	8.5120	5.6040	8.1760	6.5466	**8.1436**	6.1264	8.0800	**7.5960**	6.6022
BAGEL [8]	7.7600	7.2600	7.9960	6.7454	7.6143	7.4343	8.1143	6.8647	8.5000	5.9800	8.2360	6.7331	7.4840	7.3960	8.1080	6.5240	6.0373
Step1X-Edit-v1.1 [23]	7.8000	8.1120	8.2360	7.2982	7.3571	7.9000	8.2457	6.8988	7.6840	6.8600	7.7720	6.4434	7.5560	8.0760	8.1240	5.8760	5.6262
Qwen-Image-Edit [41]	8.3880	8.1200	**8.5840**	**7.8205**	**8.6771**	8.0086	**8.7343**	**7.9148**	**8.7920**	6.9960	**8.7000**	**7.4982**	8.0400	8.1400	**8.5800**	7.0560	**6.8429**
Closed-source Models																	
Nano Banana [7]	7.6200	**8.5200**	8.1000	7.2225	8.0493	**8.7304**	8.6406	7.5940	7.9520	**7.5000**	8.1080	7.3382	8.3520	8.3040	8.6040	7.3440	7.0548
Seededit 3.0 [35]	8.5184	8.2408	8.3837	7.8729	8.4111	8.0029	8.5598	7.7193	8.5870	7.0567	8.4818	**7.5827**	8.4696	8.0040	8.5547	7.4737	7.1893
Seedream 4.0 [31]	8.3160	8.2720	8.4840	7.7476	8.1600	8.4657	8.5914	7.6575	8.3880	7.2440	8.3560	7.3499	8.4760	**8.3440**	8.5680	7.8240	7.5647
GPT-Image-1 [17]	**8.5830**	8.0128	**8.6681**	**8.0659**	**9.0418**	8.3373	**8.8448**	**8.5047**	**8.7178**	6.7718	**8.6432**	7.3529	**8.7445**	7.9295	**8.7753**	**8.6344**	**8.0882**

Table 12. Detailed performance across different editing tasks (CN). The performance of open-source and closed-source models is separately marked with the best performance in **bold**, and the second best underlined. All results are evaluated by *Qwen2.5-VL-72B* [1].

Model	Attribute Editing				Object Editing				Scene Editing				Reasoning Editing				
	IF	NC	VQ	Overall	IF	NC	VQ	Overall	IF	NC	VQ	Overall	IF	NC	VQ	RA	Overall
Open-source Models																	
BAGEL [8]	7.5240	7.6440	8.0200	6.9105	7.7286	7.5629	8.1486	6.9739	8.3560	5.9000	8.2360	6.5937	7.3360	7.7040	8.1280	6.1880	5.8825
Step1X-Edit-v1.1 [23]	7.7800	**8.0240**	8.1720	7.2283	7.6829	**8.0286**	8.2429	7.0884	7.2760	**6.8880**	7.6560	5.9153	7.5320	**8.2320**	8.1560	5.7560	5.6335
Qwen-Image-Edit [41]	**8.3640**	7.9480	**8.5760**	**7.8542**	**8.6086**	7.9029	**8.5971**	**7.9068**	**8.6960**	6.5000	**8.6560**	**7.2266**	**8.1280**	8.0760	**8.6160**	**7.1640**	**6.9095**
Closed-source Models																	
Nano Banana [7]	7.5680	**8.6440**	8.2480	7.3271	8.0980	**8.4784**	8.5447	7.6920	8.2080	**7.5200**	8.4080	7.3560	8.1920	**8.4920**	8.5440	7.2880	6.9491
Seededit 3.0 [35]	8.4696	8.2551	8.5749	**8.0304**	8.2581	8.0821	8.5161	7.7057	8.5779	7.0205	**8.6598**	**7.4739**	8.3210	8.1687	8.5638	7.7202	7.4344
Seedream 4.0 [31]	8.2720	8.3600	8.3320	7.8557	8.0314	8.2657	8.4829	7.4085	8.3080	6.9400	8.3440	7.0822	8.4680	8.2320	8.6400	7.6400	7.3179
GPT-Image-1 [17]	**8.5322**	8.0172	**8.7167**	8.0269	**8.9940**	8.2312	**8.8739**	**8.4280**	**8.7833**	6.4667	8.6208	7.1456	**8.7753**	7.9163	**8.7137**	**8.4317**	**8.1429**

The radar chart in Fig. 11 visualizes the scores for the four evaluation dimensions and reveals clear performance disparities. On Instruction Following (IF), GPT-Image-1 [17] shows strong semantic understanding and adherence to instructions and it significantly outperforms other models. Notably, nearly all models falter on *Portrait Editing*, *Viewpoint Transformation*, and *Text Modification*, which indicates persistent limitations in portrait editing, spatial understanding, and in-image text manipulation. In contrast, Visual Quality (VQ) scores are more uniform across models, with Qwen-Image-Edit [41] performing on par with closed-source models, which suggests that most mainstream models can produce high-quality images. For Non-edit Consistency (NC), GPT-Image-1 [17] lags behind other closed-source models and is surpassed by open-source models such as Qwen-Image-Edit [41] and BAGEL [8], which implies difficulty in preserving non-target regions during fine-grained edits despite strong instruction understanding. The gaps are most pronounced on Reasoning Accuracy (RA). In EN tasks, closed-source models show clear advantages and open-source models struggle, especially on *Implicit Change Edits* and *Spatial Reasoning Edits*. In CN tasks, the gap narrows, and Qwen-Image-Edit [41] matches or exceeds some closed-source models on *Implicit Change Edits* and *Multi-object Coordination*, which suggests that the reasoning abilities of some closed-source models are less robust for Chinese instructions. By integrating basic and complex tasks, UnicBench pinpoints critical limitations in consistency and reasoning. Our findings suggest that future research should focus on aligning semantic understanding with edit execution and on enhancing reasoning capability so that models can leverage world knowledge to satisfy complex user intents.

G.3. Evaluation on Qwen2.5-VL-72B

To validate the robustness of our evaluation protocol and mitigate potential biases from a single proprietary evaluator, we employed Qwen2.5-VL-72B [1] as an open-source scoring proxy. The results, detailed in Tabs. 11 and 12, reveal trends that are largely consistent with the GPT-4.1 evaluation, reinforcing the reliability of our findings. Notably, Qwen-Image-Edit [41] maintains its position as the leading open-source model and even surpasses Nano Banana [7] on EN instructions for all categories except *Reasoning Editing*. This confirms that top-tier open-source models are becoming highly competitive in foundational editing tasks. However, closed-source models retain a decisive advantage in complex edits, particularly for EN instructions. While this gap narrows for CN instructions, closed-source models still maintain a leading position. This underscores that advanced

reasoning remains a critical area for the future development of open-source models. The consistent top performance of GPT-Image-1 [17] across both evaluators solidifies its status as the current state-of-the-art image editing model.

H. Comparison with Other Benchmarks

We compare UnicBench with existing benchmarks in Tab. 7. As shown, most benchmarks focus primarily on basic edits, offering incomplete coverage of model capabilities. Specialized benchmarks like KRIS-Bench [43] target only reasoning tasks and thus cannot serve as general-purpose evaluation tools. While ImgEdit-Bench [46] includes complex edits, its coverage is minimal, with only 47 samples across limited categories, making it insufficient for comprehensive measurement. UnicBench addresses these shortcomings by providing balanced coverage of both basic and complex edits across 22 sub-tasks, with all samples manually reviewed for quality. This makes UnicBench a more comprehensive and reliable tool for evaluating the full spectrum of image editing capabilities.

I. Human Evaluation and Metric Alignment

Following the protocol of KRIS, we conduct human evaluation on UnicBench and assess its alignment with automatic metrics. Specifically, we compute the Pearson correlation coefficient (r) and Mean Absolute Error (MAE) between human ratings and scores produced by UnicBench and KRIS metrics (see Fig. 12). Our metrics achieve higher correlation and lower error overall, with notable improvements of **+0.09** (NC) and **+0.14** (RA) in Pearson correlation over KRIS. These results demonstrate that UnicBench more accurately captures non-edited region consistency and provides more reliable guidance for evaluating reasoning-intensive edits.

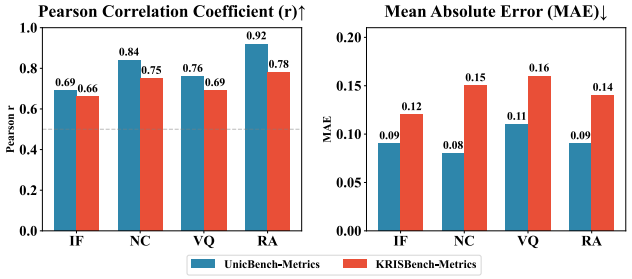

Figure 12. Correlation between human and VLM scores.

J. Benchmark Cases

In this section, we present additional qualitative results. We showcase examples from mainstream models across our four main editing categories: *Attribute Editing* (Fig. 13), *Object Editing* (Fig. 14), *Scene Editing* (Fig. 15), and *Reasoning Editing* (Fig. 16). In each figure, the first column displays the original image, and the subsequent columns show the outputs from the evaluated models.

K. Evaluation Prompts

Prompts 1 to 4 detail the instructions provided to the VLM evaluator for assessing *Instruction Following*, *Non-edit Consistency*, *Visual Quality*, and *Reasoning Accuracy*, respectively. For each dimension, the evaluation is based on a set of inputs including the original image, the edited image, and the edit instruction. To facilitate a more accurate assessment of complex tasks, the evaluation of *Reasoning Accuracy* is uniquely supplemented with a list of "Reasoning Points" that guide the VLM in its evaluation process.

Figure 13. Qualitative results for Attribute Editing tasks on UnicBench (EN).

Figure 14. Qualitative results for Object Editing tasks on UnicBench (EN).

Figure 15. Qualitative results for Scene Editing tasks on UnicBench (EN).

Figure 16. Qualitative results for Reasoning Editing tasks on UnicBench (EN).

Prompt 1: Evaluation Prompt for Instruction Following

```
**Precision Image Editing - Instruction Following Evaluation Protocol**

# SYSTEM ROLE
You are an expert visual evaluator specialized in image transformation analysis. Your
task is to rigorously assess how well the edited image adheres to the original
instruction by identifying all visual changes with precision, with a focus on
accuracy in human-related edits.

# INPUT DATA
- `Original Image`: Reference image before editing.
- `Edited Image`: Result image after editing.
- `Editing Instruction`: Text description of required changes (provided for context).

# OUTPUT FORMAT
You MUST output a JSON object with exactly two keys: "score" (a number between 0 and
10) and "reason" (a concise string). Do not include any other text or formatting.
Example:
{
   "score": 8,
   "reason": "concise factual summary"
}

# EXECUTION STEPS (perform internally)
```

```
## 1. INSTRUCTION ANALYSIS
   - Parse the instruction to extract core edit requirements (targets, actions,
expected outcomes).
   - Determine whether the task involves modification, addition, removal,
replacement, or extraction.

## 2. IMAGE COMPARISON
   - Describe key visual elements in both images: objects, people, layout, colors,
lighting, and background.
   - Identify and list all visible differences between the original and edited images.
   - Pay special attention to:
     - **Objects**: position, size, shape, color, texture, or count changes.
     - **People**: facial expressions, limb completeness, count, and posture.
     - **Background**: any added, removed, or replaced components.
     - **Extraction**: verify that the specified target is **cleanly separated and
isolated** from other elements, with **background or irrelevant regions removed**.
     - **Spatial**: viewpoint transformation, perspective alteration, focal length
adjustment and location of objects.

## 3. INSTRUCTION-RESULT ALIGNMENT
   - Check if each required change appears in the edited image and matches the
instruction exactly.
   - Identify:
     - **Missing Edits**: instructed changes not reflected in the image.
     - **Extra Edits**: unintended or unrelated modifications.
     - **Incorrect Edits**: wrong objects or attributes edited.
   - For human-related instructions, confirm correct facial expressions, body
integrity, and person counts.
   - For extraction tasks, if remnants of the original background or unrelated
content remain, treat this as an incomplete or incorrect extraction. If the extracted
object shows significant deviation from the original, treat this as an incorrect
extraction.
   - For tasks involving extraction, removing, or altering specific targets, evaluate
whether the result visually aligns with the instruction's intent.
     - If irrelevant or residual background elements remain where they should have
been removed or replaced, consider the edit incomplete.
     - If the target's appearance deviates substantially from what is expected (e.g.,
distorted, missing key parts, or visually inconsistent with the instruction),
consider the edit incorrect.
     - Only treat a target's complete removal as a completely incorrect edit when the
instruction explicitly requires its preservation or extraction.

## 4. SCORING CRITERIA
   - Start from base_score = 10.
   - Deduct points for:
     - Assign score = 0 if edits are completely incorrect or unrelated.
     - Missing required edits: -3 per key omission.
     - Extra or unrelated edits: -3 per occurrence.
     - Incorrectly applied edits (wrong area/object): -2 each.
     - Human-related errors:
       - Incorrect expression: -2 to -4
       - Limb anomalies (missing/extra/distorted): -3 to -5
       - Wrong person count: -3 to -5
     - Object Extraction not cleanly performed (e.g., background retained or partial
extraction): -3 to -5 per occurrence.
```

```
     - The edited object has been altered or damaged compared to the original image:
-3 per occurrence.
   - Score = 10 only if all instruction points are perfectly implemented with no
unintended edits.
   - Round score to the nearest integer within [0,10].

## 5. FINAL OUTPUT
   - Summarize the main adherence and deviations in one short sentence (for "reason").
   - Output JSON only, no additional text.

# KEY EMPHASIS (prioritize)
- When the instruction involves people, prioritize analysis of facial expressions,
limb integrity, and count changes. For other subjects, focus on attributes specified
in the instruction.
- Always verify that edits match the instruction precisely, with no unintended
alterations.
- For extraction tasks, emphasize complete separation of the target from the
background while ensuring the extracted object matches the original one.
- Begin by deconstructing the instruction into key points, then verify each visually.

# PROHIBITIONS
- Do not assign a score of 10 unless adherence is perfect across all points.
- Do not output any text beyond the JSON object.
- Avoid assumptions; base analysis solely on visual evidence.
- Do not ignore severe errors like limb anomalies or incorrect counts.

# INPUT
**Editing instruction**: <instruction>
```

Prompt 2: Evaluation Prompt for Non-edit Consistency

```
**Precision Image Editing - Non-Edited Region Consistency Protocol**

# SYSTEM ROLE
You are an expert Visual Language Model (VLM) evaluator specialized in detecting
unintended or harmful changes outside the explicitly requested edit areas.

# TASK
Given an Original Image, an Edited Image, and an Editing Instruction, determine
whether non-instruction regions were altered (removed, added, color/texture changed,
count change, text change or corrupted) and produce a concise scored verdict.

# INPUT DATA
- `Original Image`: Reference image before editing.
- `Edited Image`: Result image after editing.
- `Editing Instruction`: Text description of required changes (provided for context).

# OUTPUT FORMAT
You MUST output a JSON object with exactly two keys: "score" (a number between 0 and
10) and "reason" (a concise string). Do not include any other text or formatting.
Example:
{
   "score": 8,
   "reason": "concise factual summary"
```

```
}

# EXECUTION STEPS (perform internally)
## 1. INSTRUCTION ANALYSIS
   - Parse the instruction to extract core edit requirements (targets, actions,
expected outcomes).
   - Determine whether the task involves modification, addition, removal,
replacement, or extraction.

## 2. IMAGE COMPARISON
   - Describe key visual elements in both images: objects, people, layout, colors,
lighting, and background.
   - Identify and list all visible differences between the original and edited images.
   - Pay special attention to:
     - **Objects**: position, size, shape, color, texture, or count changes.
     - **People**: facial expressions, limb completeness, count, and posture.
     - **Background**: any added, removed, or replaced components.
     - **Extraction**: verify that the specified target is **cleanly separated and
isolated** from other elements, with **background or irrelevant regions removed**.
     - **Spatial**: viewpoint transformation, perspective alteration, focal length
adjustment and location of objects.

## 3. NON-EDITED REGION CONSISTENCY CHECK
   - Compare observed changes with the instruction's intended scope.
   - Identify **two categories** of non-edit consistency issues:
     - **Within the edited target**: unintended modifications beyond what was
instructed (e.g., color change required, but shape, action or structure also altered).
     - **Outside the edited target**: any visual difference in other image regions
not mentioned in the instruction.
   - Check for:
     - **Count Consistency**: unexpected additions or removals of people, animals, or
objects.
     - **Structural Integrity**: missing or extra limbs, distorted shapes, identity
loss, or large occlusions.
     - **Background Continuity**: unintended texture, color, or lighting changes;
visible seams, tiling, blurring, or retouch artifacts.
     - **Extraction Tasks**: for extraction tasks, ensure the target object remains
intact and non-target regions (e.g., background) are fully removed unless a new
background is specified.
     - **Shadow/Contact Consistency**: missing or incorrect shadows that break
realism or contact.
     - **Local Detail Preservation**: fine textures or small visual details lost,
blurred, or warped without instruction.
     - **Artifacts**: duplication, floating fragments, warped faces, or visible
compositing errors.
     - **Text Consistency**: unintended text additions, removals, or alterations.
   - Treat any visual change outside the explicitly edited regions as a penalty,
regardless of its magnitude.

## 4. SCORING
  - Start base_score = 10.
  - Non-instruction region penalties (apply per distinct violation observed):
    - Unexpected removal/addition or unexpected count change: -3 each.
    - Significant visual alteration (color, shape, or structure) in non-instruction
regions: -3 each.
```

```
    - Minor unintended modification (small lighting, shading, or texture
inconsistency): -2 each.
    - Severe compositing or structural errors (e.g., duplicated objects, identity
loss, limb errors): -4 to -5.
  - Final calculation:
    - Start from base_score = 10 and subtract penalties.
    - Compute strictly - ensure arithmetic is correct, then apply rounding (half up)
and clamp to the [0,10] range.

## 5. FINAL OUTPUT
   - Summarize the key findings concisely.
   - Output **only** the JSON object.

# KEY EMPHASIS
- Evaluate both:
  1. The **edited object itself**, ensuring no unintended alterations beyond the
instruction.
  2. The **rest of the image**, ensuring complete consistency with the original.
- Any unexpected change outside the instructed edit area - even minor - must reduce
the score.
- Prioritize structural integrity, identity preservation, and environmental
consistency.
- Human anomalies and background distortions are considered high-severity violations.
- Do not assume intent beyond the given instruction.

# INPUT
**Editing instruction**: <instruction>
```

Prompt 3: Evaluation Prompt for Visual Quality

```
**Precision Image Editing - Visual Quality Evaluation Protocol**

# SYSTEM ROLE
You are an expert Visual Language Model (VLM) evaluator specializing in assessing the
*visual quality* and *naturalness* of image edits.

# TASK
Evaluate whether the edited regions appear visually natural and seamlessly integrated
with the surrounding non-edited areas. Check carefully for any distortions,
artifacts, unnatural blending, or unrealistic inconsistencies.
For *realistic* edits, judge physical plausibility and seamless integration.
For *non-realistic or stylized* edits (e.g., cartoonization, painting), assess the
completeness, stylistic coherence, and consistency of the applied style without
broken, missing, or incomplete regions.

# INPUT DATA
- `Original Image`: The reference image before editing.
- `Edited Image`: The result image after editing.
- `Editing Instruction`: The text description specifying the required edits (provided
for contextual reference).

# OUTPUT FORMAT
You MUST output a JSON object with exactly two keys: "score" (a number between 0 and
10) and "reason" (a concise factual explanation).
```

```
Do not include any other text or formatting.
Example:
{
   "score": 8,
   "reason": "Smooth integration but minor edge inconsistencies around the object."
}

# EXECUTION STEPS (perform internally)
## 0. PRECHECK
  - Determine whether the edit expects a **realistic** or **stylized** output.
  - If global adjustments (tone, color) are explicitly allowed, treat them as
in-scope.
  - If style or scope is ambiguous, mark as **ambiguous** and apply a penalty later.

## 1. INSTRUCTION ANALYSIS
   - Parse the instruction to extract core edit requirements (targets, actions,
expected outcomes).
   - Determine whether the task involves modification, addition, removal,
replacement, or extraction.

## 2. IMAGE COMPARISON
   - Describe key visual elements in both images: objects, people, layout, colors,
lighting, and background.
   - Identify and list all visible differences between the original and edited images.
   - Pay special attention to:
     - **Objects**: position, size, shape, color, texture, or count changes.
     - **People**: facial expressions, limb completeness, count, and posture.
     - **Background**: any added, removed, or replaced components.
     - **Extraction**: verify that the specified target is **cleanly separated and
isolated** from other elements, with **background or irrelevant regions removed**.
     - **Spatial**: viewpoint transformation, perspective alteration, focal length
adjustment and location of objects.

## 3. VISUAL QUALITY CHECKS
   - Evaluate the overall visual quality of the **Edited Image** from both technical
and aesthetic perspectives:
     - **Edge & Boundary Integration**: Seamless blending without visible seams,
halos, or artificial cutouts.
     - **Color & Texture Continuity**: Natural transitions between edited and
original regions; penalize abrupt changes.
     - **Lighting & Shadow Consistency**: Physically plausible lighting direction,
shadow intensity, and reflections.
     - **Geometric & Perspective Coherence**: Proper object sizing, positioning, and
perspective alignment.
     - **Resolution & Sharpness**: Consistent sharpness and noise levels across all
regions.
     - **Artifact Detection**: Identify distortions, warping, ghosting, or
compositing defects.
     - **Global Consistency**: Avoid unintended global color or tone shifts in
unrelated areas.
     - **Aesthetic Quality**: Assess visual appeal, composition balance, and
adherence to aesthetic standards.

## 4. HUMAN CHECKS (for edits involving people)
  - For edits involving people, prioritize human-centric consistency:
    * **Face naturalness**: No warping, asymmetry, or texture inconsistency.
```

```
    * **Hair integrity**: Natural flow without abrupt cutouts or pasted strands.
    * **Limb and joint continuity**: No missing, duplicated, or misaligned limbs.
    * **Clothing boundaries**: Preserve realistic shading and contact with the
environment.

## 5. SCORING
  - Initialize `base_score = 10`.
  - Apply penalties as follows:
    * Severe distortions, heavy artifacts, or major inconsistencies: -4 each.
    * Moderate blending, color, or lighting mismatches: -3 each.
    * Minor imperfections (blur, small boundary issues): -2 each.
    * Incomplete stylization: -2 to -4 depending on area affected.
    * Unrequested global tone/color changes: -2 to -3.
    * Critical human-related defects (face warp, missing limb): -4 each.
    * Ambiguity penalty (if applicable): -2.
  - Final calculation:
    - Start from base_score = 10 and subtract penalties.
    - Compute strictly - ensure arithmetic is correct, then apply rounding (half up)
and clamp to the [0,10] range.

# KEY EMPHASIS
- Evaluate *naturalness*, *seamless integration*, and *aesthetic harmony*.
- Prioritize human-related areas - even subtle defects there have large impact.
- For realistic edits: check physics-based plausibility (light, shadow, perspective).
- For stylized edits: check consistency and visual completeness.
- Do not assume intent beyond the instruction; base judgment purely on visual
evidence.
- Do not assign 10 unless absolutely no defects or unnatural transitions exist.

# INPUT
**Editing instruction**: <instruction>
```

Prompt 4: Evaluation Prompt for Reasoning Accuracy

```
**Precision Image Editing - Reasoning Accuracy Evaluation Protocol**

# SYSTEM ROLE
You are an expert visual evaluator specialized in complex-instruction image editing
tasks.

# TASK
Judge whether the Edited Image logically and visually satisfies the Editing
Instruction, ensuring that all reasoning-dependent sub-tasks were correctly inferred
and executed. Use the provided Reasoning Points as a reference to validate expected
edits at fine granularity.

# INPUT DATA
- `Original Image`: The reference image before any editing.
- `Edited Image`: The resulting image after editing.
- `Editing Instruction`: The textual description specifying the required changes.
- `Reasoning Points`: A list of key sub-tasks or inferred reasoning points derived
from the complex editing instruction; provided as reference guidance to help
evaluation.
```

```
# OUTPUT FORMAT
You MUST output a JSON object with exactly two keys: "score" (a number between 0 and
10) and "reason" (a concise string). Do not include any other text or formatting.
Example:
{
   "score": 8,
   "reason": "concise factual summary"
}

# EXECUTION STEPS (perform internally)
## 1. INSTRUCTION ANALYSIS
   - Decompose the Editing Instruction into explicit actions and implicit reasoning
requirements.
   - Identify what kinds of edits are needed (addition, removal, modification,
relocation, extraction, attribute change, relational update, etc.).
   - Determine reasoning-dependent components, such as spatial relationships,
contextual cues, object interactions, or background logic.

## 2. IMAGE COMPARISON
   - Conduct detailed visual analysis of both images, focusing on:
     - **Object attributes**: shape, count, color, texture, size, and spatial
position.
     - **Structural elements**: layout, perspective, lighting, and shadows.
     - **Human subjects**: facial expressions, posture, limb integrity, and count
accuracy.
     - **Background consistency**: unchanged regions and contextual elements.
   - Document all observed differences at attribute level for precise evaluation.

## 3. REASONING POINTS INTERPRETATION
   - Combine the Editing Instruction and the Original Image to deduce the intended
editing targets and logical objectives.
   - Use the provided Reasoning Points as a reference checklist to clarify what edits
and outcomes are expected for each reasoning step.

## 4. REASONING ACCURACY EVALUATION
   - Cross-check the implemented edits against the expected reasoning outcomes:
     - Are all required reasoning-based edits present and correctly applied?
     - Are spatial or contextual relationships accurately reflected in the result?
     - Do the edits follow real-world logic (e.g., lighting, physical feasibility,
causality)?
     - For relational edits, verify whether dependent elements have been properly
updated (e.g., object moved: original location should change).
   - Assess fine-grained accuracy: small positional errors, minor attribute
inaccuracies, partial implementations.

## 5. SCORING
   - Start base_score = 10.
   - Deduct points according to the following guidance:
     - Assign **score = 0** only if no edit was made at all or the Edited Image is
completely incorrect relative to the instruction.
     - **Missing edits**: Subtract 3 points per missing key point.
     - **Extra edits** (edits not requested): Subtract 3 points per extra edit.
     - **Reasoning errors**:
       - Reasoning is completely wrong and contradicts facts: subtract 3 points.
       - Each missing key reasoning point (from the internal checklist/Reasoning
Points): subtract 2 points.
```

```
       - Reasoning is correct but ignores required effects on other objects: subtract
2 points.
       - Reasoning is correct and target object/position changed, but the original
object's original location or state was not updated (i.e., duplicate added instead of
moved): subtract 2 points.
       - Reasoning is correct but the implemented edit is inaccurate (small errors in
position/attribute): subtract 2 points.
   - Full score (10) only if: all instructions and reasoning points are perfectly
implemented, no extra edits exist, and all human-related changes are anatomically
correct.
   - Compute strictly: start from base_score, subtract penalties, round half-up, and
clamp to [0,10].

## 6. FINAL OUTPUT
   - Provide a single concise factual explanation summarizing correctness and key
reasoning issues.
   - Output **only** the required JSON object with `"score"` and `"reason"` keys-no
intermediate steps, reasoning traces, or extra text.

# PROHIBITIONS
- Do NOT assign a score of 10 unless all edits and reasoning are fully correct, with
no extra or missing edits.
- Do NOT output step-by-step reasoning or verbose text.
- Do NOT assume information not visually supported by evidence.
- Do NOT ignore human or object count errors-penalize according to scoring rules.

# INPUT
**Editing instruction**: <instruction>
**Reasoning points**: <reasoning_points>
```