MTEDIT: PACKED TRAJECTORIES BEAT ROLLOUTS FOR MULTI-TURN IMAGE EDITING

Anonymous authors
Paper under double-blind review

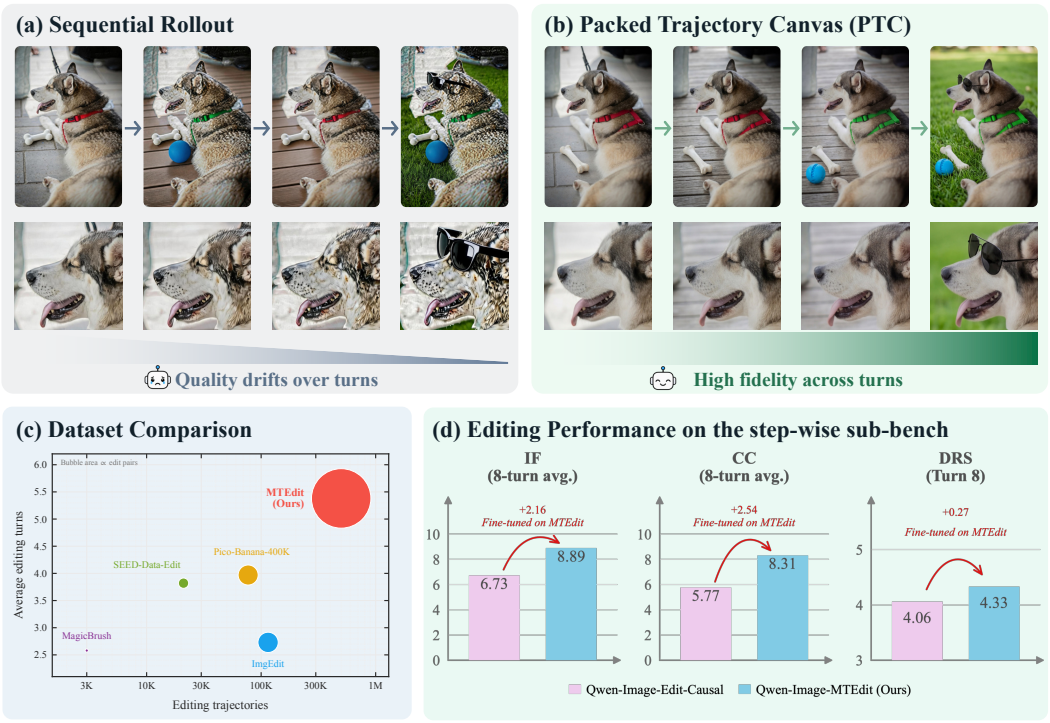


Figure 1: **Overview of our approach.** (a,b) Packed trajectory canvas (PTC) generation reduces visual-quality drift compared with sequential rollout. (c) MTEdit offers more trajectories and more turns than existing datasets. (d) Fine-tuning on MTEdit improves the model's multi-turn editing performance in instruction following (IF), content consistency (CC), and visual-quality stability (DRS).

ABSTRACT

Multi-turn image editing aims to modify an image according to a sequence of instructions, making it better aligned with human-AI collaborative creation than single-turn editing. However, constructing high-quality multi-turn training data remains challenging, as conventional sequential rollout accumulates errors and introduces visual-quality drift across editing turns. To address this challenge, we introduce **MTEdit**, a multi-turn image editing dataset built through a pipeline centered on **packed trajectory canvas (PTC)** generation. By jointly generating multiple editing states within a unified visual context, this pipeline reduces visual-quality drift across turns. Covering both step-wise and context-aware editing scenarios, MTEdit comprises 500K multi-turn image editing trajectories, with up to 8 turns and an average of 5.38 turns per trajectory, making it one of the most extensive datasets for multi-turn image editing. We further introduce **MTEdit-Bench**, a benchmark designed to evaluate both scenarios by assessing models' ability to follow editing instructions, maintain content consistency, resist visual-quality drift, and effectively use historical context across turns. Using MTEdit, we fine-tune Qwen-Image-Edit-Causal to obtain **Qwen-Image-MTEdit**. On MTEdit-Bench, we evaluate a broad range of closed-

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.

source and open-source image editing models, including Qwen-Image-MTEdit. Experimental results show that Qwen-Image-MTEdit achieves the best performance among the evaluated open-source models, demonstrating the effectiveness of MTEdit in improving models' multi-turn image editing capabilities. Code and data will be publicly available upon acceptance. For additional results, see `https://sadlksjada.github.io/mtedit.github.io/`.

1 INTRODUCTION

Recently, multi-turn image editing has attracted increasing attention from the research community. This setting can be broadly categorized into two types: 1) *step-wise* editing Zhou et al. (2025); Liao et al. (2025); Ye et al. (2026b;a), where each editing step is performed independently based on the previous output image and the current instruction; and 2) *context-aware* editing Ye et al. (2025b); Qu et al. (2026); Deng et al. (2025); Zhang et al. (2026a), where the model generates the target image by jointly considering previous text instructions and historically generated images within a continuous editing context. Compared with conventional single-turn editing Brooks et al. (2023); Sheynin et al. (2024); Huang et al. (2024); Li et al. (2025); Mao et al. (2025); Cui et al. (2025); Zhang et al. (2025), multi-turn image editing better aligns with the iterative workflow of real-world creative scenarios, suggesting substantial potential for practical applications.

Realizing this potential requires high-quality training data for both editing scenarios. Sequential rollout with powerful closed-source image editing models is the most common approach to constructing such data, as it combines single-turn edits with a scalable generation process Qian et al. (2025). However, this approach reuses each generated image as the input to the next edit, allowing errors and artifacts to accumulate and causing visual-quality drift over longer trajectories. Beyond visual quality, existing datasets are also limited in trajectory length and often cover only one of the two editing scenarios Zhang et al. (2023); Ye et al. (2025b); Qu et al. (2026).

To address these challenges, we introduce **MTEdit**, a multi-turn image editing dataset constructed through a pipeline centered on **packed trajectory canvas (PTC)** generation. A PTC contains multiple panels, each depicting the outcome of one editing turn. Jointly generating these panels within a unified visual context reduces visual-quality drift across turns, enabling the construction of high-quality multi-turn trajectories. Specifically, given a curated source image, a vision-language model (VLM) plans an editing sequence and produces a corresponding *PTC prompt*. The source image and this prompt are then provided to a closed-source image editing model (e.g., GPT Image 2 OpenAI (2026) or Nano Banana 2 Google (2026)) to generate the PTC. Gemini 3.1 Pro Google DeepMind (2026) then filters out failed PTC generations before deriving turn-level editing instructions from each retained PTC and its source image. The retained canvases are subsequently split into target images to form training pairs. Finally, we apply SIFT-based Brown & Lowe (2007) feature matching followed by image warping to improve pixel-level alignment within each editing pair.

MTEdit comprises a total of 500K multi-turn editing trajectories and 2.69M edit pairs, with 4 to 8 turns per trajectory and an average of 5.38 turns. It covers both step-wise and context-aware editing scenarios and supports 23 common editing categories (see Fig. 3), making it an extensive resource for training multi-turn image editing models. As shown in Tab. 1 and Fig. 1(c), MTEdit offers more trajectories and longer editing sequences than the compared datasets. Formal definitions of step-wise and context-aware editing are provided in Sec. 3.1.

Beyond training data, reliable evaluation is essential for understanding model performance across successive editing turns. Although existing benchmarks Qu et al. (2026); Chen et al. (2025b); Ye et al. (2026b); Xu et al. (2026) have begun to assess multi-turn image editing, many use sequences of only around three turns Ye et al. (2025b); Zhang et al. (2023); Ye et al. (2026b); Chen et al. (2025b), limiting their ability to reveal error accumulation over longer trajectories. Their evaluation protocols also often overlook visual-quality drift across turns. To address these limitations, we introduce **MTEdit-Bench** for evaluating both step-wise and context-aware editing. It assesses instruction following, content consistency, resistance to visual-quality drift, and the ability to use historical instructions and image states throughout the editing process.

To demonstrate the training utility of MTEdit, we fine-tune Qwen-Image-Edit-Causal ModelTC (2026) on the dataset to obtain **Qwen-Image-MTEdit**. We evaluate a broad range of closed-source and open-source image editing models on MTEdit-Bench, including Qwen-Image-MTEdit. Our

Table 1: **Comparison of representative multi-turn image editing datasets.** #Traj. and #Edits denote the numbers of editing trajectories and edit pairs; Max T. and Avg T. denote the maximum and average numbers of turns; Step-wise and Context-aware indicate whether the dataset supports the two multi-turn editing settings; Construction describes how the dataset is built.

Dataset	#Traj.	#Edits	Max T.	Avg T.	Step-wise	Context-aware	Construction
MagicBrush Zhang et al. (2023)	3K	8K	3	2.58	✓	✗	Human
SEED-Data-Edit Ge et al. (2024)	21K	81K	6	3.82	✓	✗	Human
ImgEdit Ye et al. (2025b)	115K	314K	3	2.73	✗	✓	Inpainting
Pico-Banana-400K Qian et al. (2025)	77K	306K	6	3.97	✓	✗	Sequential rollout
MTEdit (Ours)	**500K**	**2.69M**	**8**	**5.38**	✓	✓	**Packed trajectory canvas**

model achieves the best performance among the evaluated open-source models, demonstrating the effectiveness of MTEdit in improving multi-turn image editing capabilities.

Our main contributions are summarized as follows: 1) We introduce **MTEdit**, comprising 500K multi-turn editing trajectories constructed through a **PTC** generation pipeline that reduces visual-quality drift across turns; 2) We introduce **MTEdit-Bench** to evaluate both step-wise and context-aware editing in terms of instruction following, content consistency, visual-quality stability, and historical context use; 3) We develop **Qwen-Image-MTEdit** by fine-tuning on MTEdit, achieving the best performance among the evaluated open-source models on MTEdit-Bench.

2 Related Work

Instruction-based Image Editing: Instruction-based image editing (IIE) modifies an input image according to a natural language command while preserving instruction-irrelevant visual content. InstructPix2Pix Brooks et al. (2023) is the first work to formalize this paradigm by leveraging Prompt-to-Prompt Hertz et al. (2022) to ensure spatial-structure consistency in training data. Recent closed-source models such as the GPT-Image series OpenAI (2025a;b; 2026) and Nano Banana series Google (2025a;b; 2026), together with open-source models such as Step1X-Edit Liu et al. (2025), FLUX Labs et al. (2025b), Qwen-Image-Edit Wu et al. (2025a), LongCat-Image Team et al. (2025), FireRed-Image-Edit Team (2026b), and JoyAI-Image Song et al. (2026), have continued to advance the field. The progress in this field largely depends on large-scale, high-quality training data Meng et al. (2021); Sheynin et al. (2024); Bai et al. (2024); Fu et al. (2024); Zhang et al. (2024); Chen et al. (2025d); Zhang et al. (2026b); Chen et al. (2026). However, most efforts still focus on single-turn editing, leaving the more interactive multi-turn setting comparatively underexplored.

Datasets for Single-turn Image Editing: Most existing image editing datasets are constructed under the single-turn editing setting. Early datasets such as GIER Shi et al. (2021) provide human-annotated editing instructions with paired before-and-after images, establishing a standard supervision format for instruction-guided image editing. Subsequent datasets scale this paradigm through synthetic or hybrid construction pipelines, including UltraEdit Zhao et al. (2024), OmniEdit Wei et al. (2025), AnyEdit Yu et al. (2025), and UniVG Fu et al. (2025), while frontier-model-distilled datasets such as Echo-4o-Image Ye et al. (2025a), HQ-Edit Hui et al. (2024), GPT-Image-Edit-1.5M Wang et al. (2025) and ShareGPT-4o-Image Chen et al. (2025a) further expand the scale and complexity of image editing supervision. Despite their effectiveness, these datasets mainly supervise one-step transformations, motivating dedicated datasets that capture coherent editing sequences across multiple successive user interaction turns rather than isolated before-and-after image pairs.

Datasets for Multi-turn Image Editing: MagicBrush Zhang et al. (2023) provides early multi-turn samples for real images, but its trajectories are short, small-scale, and heavily human-annotated. SEED-Data-Edit Ge et al. (2024) introduces Photoshop-expert annotations, yet its multi-turn subset remains limited and lacks systematic coverage of context-dependent editing. ImgEdit Ye et al. (2025b) explores context-aware abilities such as content understanding, content memory, and version backtracking, but it has short editing trajectories and overlooks common step-wise editing data. Pico-Banana-400K Qian et al. (2025) constructs about 77K multi-turn sequences using Nano Banana, but its rollout pipeline can accumulate errors and introduce visual drift in longer trajectories. VINCIE Qu et al. (2026) extracts interleaved image-text sequences from videos for in-context editing, but natural video transitions are insufficient to cover diverse user editing intents. Overall, high-quality, comprehensive, and large-scale datasets for multi-turn image editing remain lacking.

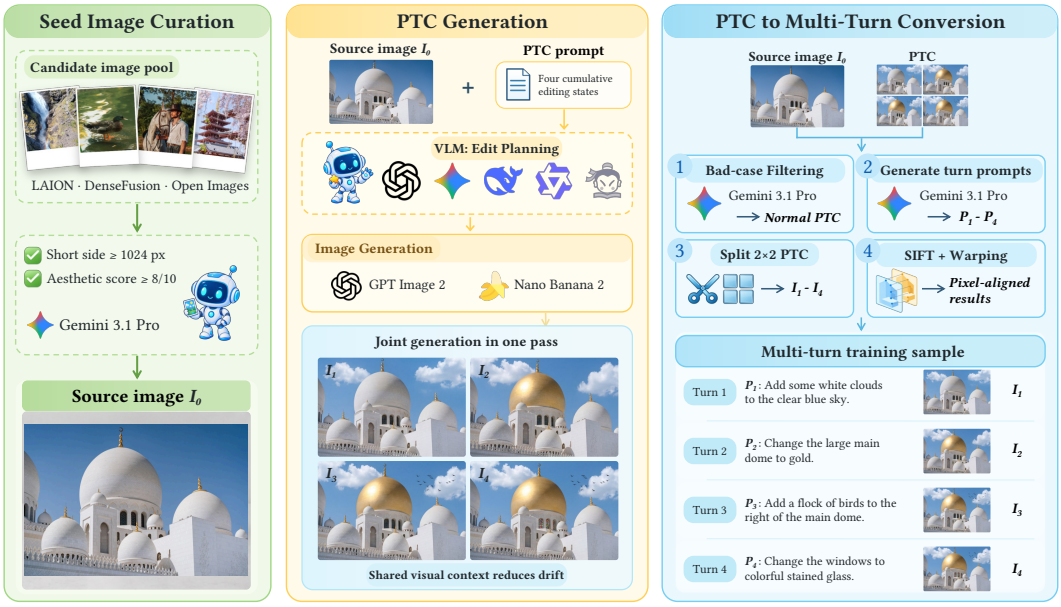


Figure 2: **Overview of our data construction pipeline.** (a) Seed image curation, which selects source images with high resolution and aesthetic quality; (b) PTC generation, in which VLMs derive a PTC prompt from each source image to guide closed-source image models in generating the corresponding PTC; and (c) PTC to multi-turn conversion, which processes the generated canvases into multi-turn editing trajectories with turn-level instructions and target images.

3 Technical Details

This section presents the technical details of MTEdit, MTEdit-Bench, and Qwen-Image-MTEdit. Sec. 3.1 first defines the tasks covered by our dataset: *step-wise* editing and three types of *context-aware* editing. Sec. 3.2 details the data construction pipeline of MTEdit. Sec. 3.3 analyzes the scale of MTEdit, the distribution of editing types, and its coverage of multi-turn scenarios. Sec. 3.4 introduces the construction of MTEdit-Bench and its automated evaluation protocol. Sec. 3.5 further describes Qwen-Image-MTEdit, which is fine-tuned on MTEdit.

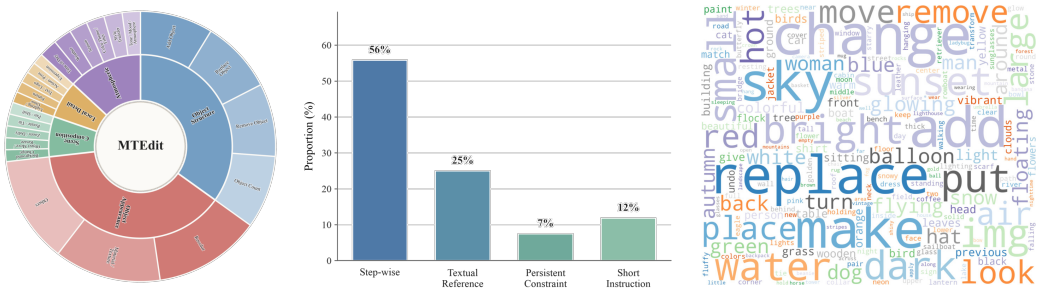
3.1 Task Definition

Step-wise Editing is the most common setting in multi-turn image editing and is also the primary form studied in most prior work Zhou et al. (2025); Liao et al. (2025); Ye et al. (2026b). In this setting, each editing turn does not explicitly rely on the historical context composed of all previous images and text instructions. Instead, it is mainly conditioned on the image generated in the previous turn and the current user edit prompt. Therefore, models designed for single-turn editing can also be extended to step-wise editing through rollout-style execution. However, for single-turn edit models, this strategy causes errors to accumulate across editing turns. As the number of turns increases, image content, subject consistency, and local details often exhibit noticeable drift Tang et al. (2026).

Context-aware Editing requires the model to leverage information from previous turns, rather than relying solely on the latest image and current instruction. In this paper, we focus on three types of context-aware editing: (1) *persistent-constraint* editing, where constraints specified in earlier prompts must remain valid in subsequent turns; (2) *textual-reference* editing, where later instructions refer to previously introduced or modified objects through pronouns or descriptive phrases; and (3) *short-instruction* editing, where an initial explicit instruction establishes the edit target and later-turn instructions are brief or underspecified, often requesting adjustments to the degree or intensity of an attribute.

3.2 Data Construction Pipeline

The data construction pipeline of MTEdit consists of three steps: (a) seed image curation, (b) PTC generation, and (c) PTC to multi-turn conversion, as illustrated in Fig. 2.



(a) Edit Type Taxonomy (b) Multi-turn Data Distribution (c) Instruction Word Cloud

Figure 3: **Statistics of MTEdit.** In this figure, we present MTEdit's (a) edit-type taxonomy, (b) multi-turn data distribution, and (c) instruction word cloud.

Seed Image Curation: To construct multi-turn editing trajectories, we first curate seed images that serve as the source images of the trajectories. We collect candidate images from large-scale, openly available image sources Chen et al. (2025c), including LAION Schuhmann et al. (2022), Open Images Kuznetsova et al. (2020) and DenseFusion Li et al. (2024). Each retained seed image is used as the initial state of one multi-turn editing trajectory.

To ensure that the source images have high visual quality, we apply two filtering criteria: (i) high resolution, requiring the shorter side of the image to be at least 1024 pixels; and (ii) high aesthetic quality, requiring an aesthetic score of at least 8 on a 0–10 scale, as assessed by Gemini 3.1 Pro Google DeepMind (2026). The system prompt used for this can be found in Appendix A.8.1.

PTC Generation: A common strategy Qian et al. (2025) for constructing multi-turn image editing trajectories is to leverage the capabilities of closed-source image editing models and generate data in a sequential rollout manner, where the output of each editing turn is used as the input to the next turn. However, the major limitation of this strategy is that visual quality drift becomes increasingly severe as the number of editing turns grows (e.g., beyond four turns).

To address this issue, we use closed-source models to generate multiple editing states within a single PTC in one pass. By jointly generating targets for multiple editing turns in a shared context, this approach reduces visual quality drift and enables longer editing trajectories with better visual quality. Qualitative and quantitative comparisons with rollout-style pipelines are presented in Fig. 8 and Tab. 4 in the appendix, respectively. PTC generation proceeds in two steps. **First**, we use VLMs to synthesize an instruction for closed-source models from the source image, named a *PTC prompt*. **Second**, we feed the source image and *PTC prompt* to closed-source models to obtain the PTC. The generated PTCs are 4K square canvases with four panels in a 2×2 layout, ensuring that each panel retains a resolution of approximately 2K, which is sufficient for producing high-quality training data.

PTC to Multi-Turn Conversion: After obtaining the PTC, we convert it into multi-turn image editing trajectories. This conversion consists of four steps. **First**, we use Gemini 3.1 Pro to filter out failed PTC generations, such as those shown in Fig. 13. **Second**, we feed the source image with the retained PTC into Gemini 3.1 Pro to generate user prompts for each editing turn. **Third**, we split the PTC into panels according to the 2×2 layout, and treat each panel as the target image after the corresponding editing turn. **Fourth**, we apply SIFT-based Brown & Lowe (2007) feature matching followed by image warping to improve pixel-level alignment within each editing pair. As a result, each sample consists of the source image, turn-level user prompts, and turn-level target images.

Moreover, under the default 2×2 PTC layout, each run of our data construction pipeline produces a single four-turn editing sample. To construct longer editing sequences (such as eight-turn samples), we only need to take the generated image from the last turn as the new source image for the next round and then run the same data construction pipeline again.

3.3 Dataset Statistics

MTEdit contains 500K editing trajectories and 2.69M edit pairs, with up to 8 turns and an average of 5.38 turns per trajectory, and simultaneously covers both step-wise and context-aware multi-turn editing. In terms of the number of turns, the total number of edit pairs, and scenario completeness, our dataset provides broader coverage than the compared multi-turn editing datasets.

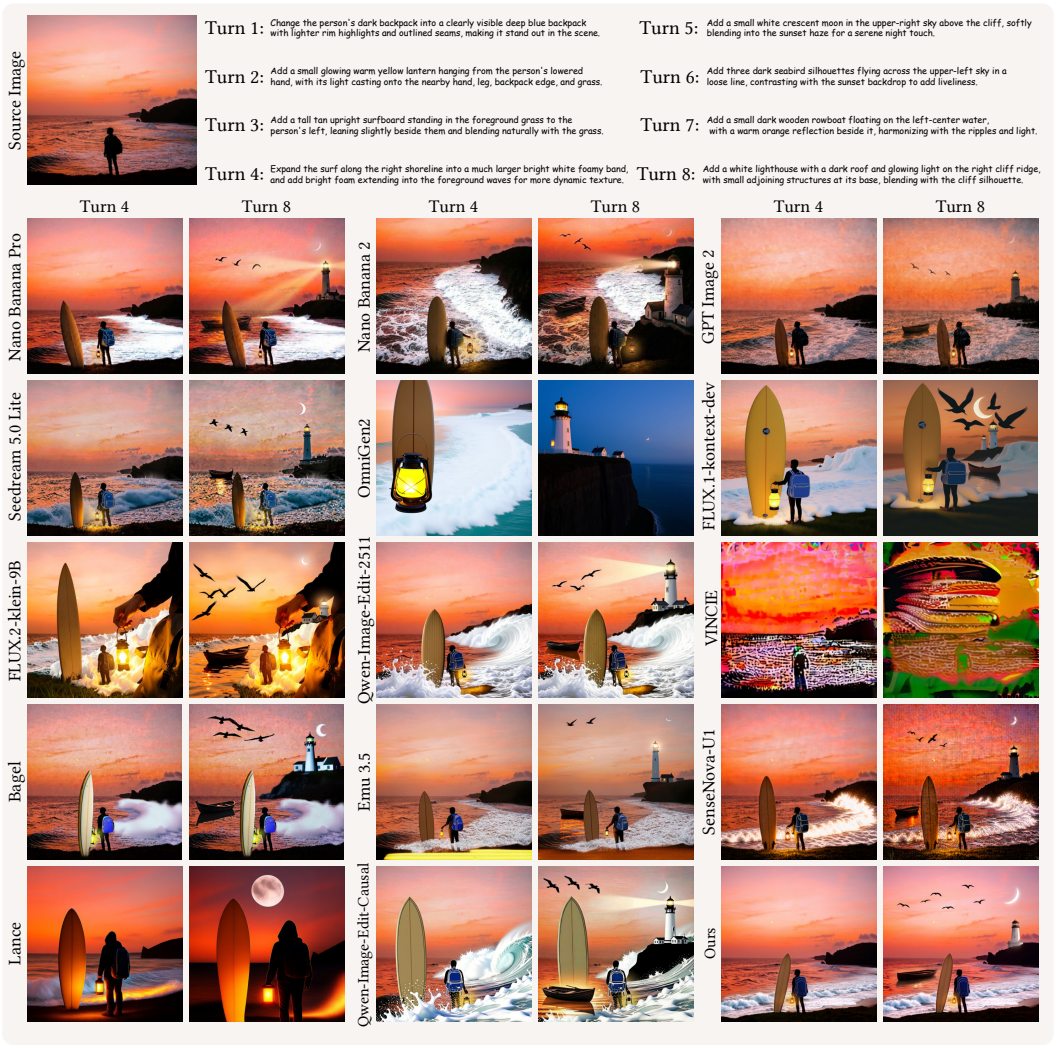


Figure 4: **Qualitative results on the step-wise sub-bench.** Our method achieves competitive editing performance and maintains visual quality without noticeable degradation across editing turns.

Fig. 3 further characterizes the distribution of MTEdit from three perspectives. The left subfigure summarizes the edit-type taxonomy used in MTEdit. We organize all edits into a two-level hierarchy, with five coarse categories: object structure, object appearance, scene composition, local detail, and atmospheric edits. These coarse categories are further divided into 23 leaf categories. The middle subfigure reports the proportion of each multi-turn data type: step-wise data accounts for 56%, while the remaining 44% are context-aware data, further divided into textual-reference (25%), persistent-constraint (7%) and short-instruction (12%). The right subfigure presents a word cloud of all user instructions in MTEdit, illustrating its lexical diversity.

3.4 MTEDIT-BENCH

Benchmark Construction and Task Coverage: MTEdit-Bench comprises source images paired with manually crafted prompts, covering four multi-turn editing tasks: step-wise, persistent-constraint, textual-reference, and short-instruction. The step-wise sub-bench consists of 100 eight-turn samples and mainly evaluates editing performance and visual drift over long editing chains. Each of the three context-aware sub-benches consists of 50 four-turn samples and additionally considers models' ability to leverage prior textual prompts and visual states across turns.

Evaluation Metrics: MTEdit-Bench uses task-specific metrics to evaluate capabilities required across multi-turn editing scenarios. For the step-wise sub-bench, we report three turn-level metrics: IF (Instruction Following), CC (Content Consistency), and DRS (Drift Robustness Score).

Table 2: **Quantitative results on the step-wise sub-bench.** Closed-source scores are shown in gray; among open-source methods and our model, different cell colors refer to best and 2nd-best results. For all metrics, higher values indicate better performance.

Method	Turn 1			Turn 2			Turn 3			Turn 4		
	IF	CC	DRS	IF	CC	DRS	IF	CC	DRS	IF	CC	DRS
Nano Banana Pro Google (2025b)	9.75	9.79	4.17	9.73	9.53	4.22	9.63	9.35	4.22	9.53	9.06	4.19
Nano Banana 2 Google (2026)	9.65	9.58	4.19	9.45	9.21	4.21	9.28	8.85	4.23	9.31	8.53	4.20
GPT Image 2 OpenAI (2026)	9.02	8.72	4.22	8.66	8.62	4.23	8.65	8.07	4.22	8.62	7.82	4.20
Seedream 5.0 Lite ByteDance Seed (2026)	9.73	9.60	4.27	9.81	9.08	4.29	9.66	8.57	4.26	9.34	7.69	4.19
VINCIE Qu et al. (2026)	1.13	4.31	2.47	0.01	0.02	1.66	0.00	0.00	1.40	0.00	0.00	1.33
OmniGen2 Wu et al. (2025b)	6.75	7.46	4.00	4.44	4.95	3.77	3.02	2.75	3.43	1.52	1.24	3.05
FLUX.1-kontext-dev Labs et al. (2025b)	8.29	8.77	4.12	7.41	7.61	4.12	6.40	6.42	4.12	5.04	5.08	4.09
FLUX.2-klein-9B Labs (2025)	9.09	9.19	4.15	8.63	8.83	4.17	8.23	8.03	4.21	8.06	7.53	4.20
Qwen-Image-Edit-2511 Wu et al. (2025a)	9.30	9.43	4.12	9.07	8.87	4.15	8.67	8.11	4.16	8.31	7.51	4.16
BAGEL Deng et al. (2025)	8.69	9.14	3.99	7.56	8.18	3.92	6.48	6.95	3.88	4.57	5.55	3.71
Emu3.5 Cui et al. (2025)	7.40	7.90	4.03	7.09	6.83	4.01	6.17	5.98	4.16	5.69	5.30	4.13
SenseNova-U1 Diao et al. (2026)	9.29	9.06	4.07	8.95	8.51	4.14	8.53	7.61	4.12	7.37	5.95	3.90
Lance Fu et al. (2026)	8.20	8.39	4.18	6.78	6.25	4.29	4.44	3.77	4.32	2.59	1.78	4.30
Qwen-Image-Edit-Causal ModelTC (2026)	8.89	9.21	4.18	8.76	8.21	4.22	8.54	7.26	4.20	7.75	6.38	4.17
Ours	9.74	9.54	4.23	9.59	9.28	4.26	9.32	9.19	4.30	9.35	8.82	4.33

Method	Turn 5			Turn 6			Turn 7			Turn 8		
	IF	CC	DRS	IF	CC	DRS	IF	CC	DRS	IF	CC	DRS
Nano Banana Pro Google (2025b)	9.54	8.73	4.15	9.45	8.49	4.12	8.94	8.31	4.08	8.83	7.51	4.05
Nano Banana 2 Google (2026)	9.24	8.38	4.19	8.98	7.86	4.19	8.34	7.44	4.18	7.99	6.80	4.20
GPT Image 2 OpenAI (2026)	8.58	7.32	4.18	8.23	7.00	4.16	7.41	6.26	4.17	7.11	5.81	4.17
Seedream 5.0 Lite ByteDance Seed (2026)	9.07	7.11	4.10	8.89	6.33	4.00	7.31	5.33	3.88	6.38	4.52	3.75
VINCIE Qu et al. (2026)	0.00	0.00	1.31	0.00	0.00	1.38	0.00	0.00	1.39	0.00	0.00	1.38
OmniGen2 Wu et al. (2025b)	0.58	0.48	2.98	0.31	0.21	2.95	0.20	0.07	2.98	0.00	0.00	3.10
FLUX.1-kontext-dev Labs et al. (2025b)	4.38	4.04	4.00	3.62	3.20	3.95	2.47	2.50	3.91	2.13	1.76	3.82
FLUX.2-klein-9B Labs (2025)	7.48	7.12	4.20	7.40	6.58	4.17	6.39	6.23	4.18	6.07	5.43	4.12
Qwen-Image-Edit-2511 Wu et al. (2025a)	8.15	7.06	4.13	7.56	6.52	4.11	7.10	6.15	4.12	6.87	5.56	4.05
BAGEL Deng et al. (2025)	3.99	4.14	3.62	2.79	3.19	3.37	2.15	2.17	3.24	1.60	1.68	3.06
Emu3.5 Cui et al. (2025)	5.15	4.74	4.13	4.40	4.21	4.17	3.48	3.17	4.06	3.01	2.72	4.12
SenseNova-U1 Diao et al. (2026)	6.21	4.41	3.55	4.50	3.06	3.14	2.96	1.96	2.84	1.67	1.12	2.58
Lance Fu et al. (2026)	0.66	0.58	4.28	0.18	0.13	4.24	0.00	0.03	4.20	0.00	0.00	4.11
Qwen-Image-Edit-Causal ModelTC (2026)	6.87	5.20	4.15	5.74	4.29	4.13	4.11	3.21	4.12	3.17	2.41	4.06
Ours	8.99	8.12	4.32	8.70	7.60	4.33	8.00	7.51	4.35	7.45	6.42	4.33

IF and CC follow the VLM-judge-based evaluation paradigm used in VIEScore Ku et al. (2024), GEdit-Bench Liu et al. (2025), Edit-R2 Ye et al. (2026b), and EdiVal-Agent Chen et al. (2025b), measuring instruction following and content consistency. To compute IF at each turn, we provide the VLM with the image before the current edit, the model's generated image after the edit, and the current-turn instruction. To compute CC, we provide the initial image of the editing sequence, the model's generated image at the current turn, and all instructions up to that turn. The VLM assigns each metric a score from 0 to 10 (see Appendix A.8 for the system prompts). If either IF or CC falls below 6 at any turn, we consider the editing trajectory to have failed and set both scores to 0 for all subsequent turns. This reflects the causal nature of multi-turn editing: subsequent edits build on an incorrect state after a failure and are therefore no longer credited as successful continuation, consistent with the failure-propagation principle used in VINCIE Qu et al. (2026). To balance evaluation performance and reproducibility, we use Qwen 3.5 Plus Team (2026a) as the VLM evaluator.

DRS measures robustness against visual drift and is computed from two drift-sensitive metrics, VisualQuality-R1 Wu et al. (2025c) and RALI Zhao et al. (2025). DRS is calculated as the average of their image quality scores, $(\text{VisualQuality-R1} + \text{RALI})/2$, and ranges from 0 to 5. This choice is motivated by Banana100 Tang et al. (2026), which finds that VisualQuality-R1 and RALI

Table 3: **Quantitative ablation results on the step-wise sub-bench.** Method denotes the strategy used to construct the training data: PTC generation or sequential rollout. The best results are **bolded**. Higher values indicate better performance for all metrics.

	Turn 1			Turn 2			Turn 3			Turn 4			Turn 5			Turn 6			Turn 7			Turn 8		
Method	IF	CC	DRS	IF	CC	DRS	IF	CC	DRS	IF	CC	DRS	IF	CC	DRS	IF	CC	DRS	IF	CC	DRS	IF	CC	DRS
PTC	9.64	**9.61**	4.17	**9.63**	**9.27**	4.26	**9.30**	**8.59**	**4.21**	**8.67**	**7.67**	**4.22**	**8.21**	**7.15**	**4.21**	**7.63**	**6.47**	**4.21**	**6.24**	**5.37**	**4.23**	**5.06**	**4.44**	**4.22**
Sequential rollout	**9.66**	9.50	**4.19**	9.32	9.04	**4.27**	8.97	8.20	4.18	7.95	7.58	4.19	7.66	6.66	4.16	4.21	6.16	4.03	4.92	4.97	4.03	4.26	3.84	3.96

reliably detect visual quality degradation during iterative image replication, assigning lower scores to degraded images. We use DRS to evaluate all types of multi-turn editing data.

For the remaining three types of context-aware editing, it is necessary to consider whether models can correctly use the historical context. We therefore provide the VLM judge with the full multi-turn editing trajectory, including the instructions and corresponding generated images, allowing it to assess each case within the shared interaction context. We denote the resulting instruction-following and content-consistency scores as IF^* and CC^*, respectively. The same failure-propagation rule applies to these two scores; DRS is never zeroed after a failure.

3.5 QWEN-IMAGE-MTEDIT

To assess the training utility of MTEdit, we fine-tune Qwen-Image-Edit-Causal ModelTC (2026) on this data and obtain **Qwen-Image-MTEdit**. Building on the block causal attention, we adopt a teacher-forcing training strategy Chen et al. (2024), which allows multiple editing turns within the same trajectory to be trained jointly rather than through sequential rollout. In addition, the model uses 3D interleaved RoPE to encode the temporal order of interleaved image-text units, enabling it to distinguish different editing turns and align each instruction with its corresponding image state. Further analysis and detailed training settings are provided in Appendix A.1.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Benchmark and Metrics: The construction details and evaluation protocol of MTEdit-Bench can be found in Sec. 3.4. In brief, the step-wise sub-bench reports IF, CC, and DRS; the remaining three context-aware sub-benches report IF^*, CC^*, and DRS. All tables present turn-wise results, where each score at a turn is averaged over all test samples evaluated at that turn.

Compared Methods: We group the compared methods into two categories based on their ability to use historical image-text context. The first category includes Nano Banana Pro Google (2025b), Nano Banana 2 Google (2026), GPT Image 2 OpenAI (2026), Seedream 5.0 Lite ByteDance Seed (2026), BAGEL Deng et al. (2025), Emu3.5 Cui et al. (2025), SenseNova-U1 Diao et al. (2026), Lance Fu et al. (2026), VINCIE Qu et al. (2026), and our Qwen-Image-MTEdit. These methods can leverage historical images and instructions, supporting both step-wise and context-aware multi-turn editing. The second category includes OmniGen2 Wu et al. (2025b), FLUX.1-kontext-dev Labs et al. (2025a), FLUX.2-klein-9B Labs (2025), Qwen-Image-Edit-2511 Wu et al. (2025a), and Qwen-Image-Edit-Causal ModelTC (2026). These methods perform multi-turn editing by conditioning each turn on the previous output image and the current instruction. Since they do not explicitly model historical context, we evaluate them only in the step-wise setting.

4.2 EXPERIMENTAL RESULTS

In this subsection, we mainly present and analyze the quantitative and qualitative results of different methods on the step-wise sub-bench. Detailed analyses of the persistent-constraint, textual-reference, and short-instruction sub-benches are provided in the Appendix A.3.

Quantitative Results: Tab. 2 presents quantitative comparisons on the step-wise sub-bench. Qwen-Image-MTEdit demonstrates strong resistance to visual drift over long editing sequences: it achieves a DRS of 4.33 at Turn 4 and maintains the highest DRS among compared methods through Turn 8. Although it trails the strongest closed-source models in instruction following and content preservation, it achieves state-of-the-art performance among the compared open-source methods. Compared with our base model, Qwen-Image-Edit-Causal ModelTC (2026), Qwen-Image-MTEdit



Figure 5: **Visual quality comparison at Turn 8.** From left to right: the source image and the Turn 8 outputs from models trained on data from our PTC generation pipeline and sequential rollout data construction. Enlarged crops show fewer artifacts and cleaner details with our pipeline.

consistently improves IF, CC, and DRS across all eight turns. These results demonstrate the effectiveness of high-quality training data in improving step-wise multi-turn editing performance.

Qualitative Results: Fig. 4 presents qualitative comparisons of different models at Turns 4 and 8 of the eight-turn step-wise editing task. The compared methods exhibit two main failure modes. First, methods such as OmniGen2, the FLUX series, SenseNova-U1, and VINCIE struggle to preserve content consistency, producing outputs at Turn 8 that differ substantially from the source image beyond the intended edits. Second, most methods, including closed-source models, exhibit increasing visual quality drift as editing progresses. For example, noticeable artifacts appear in the sky at Turn 8. In comparison, our model better preserves visual quality across editing turns and produces no obvious artifacts in the final turn.

4.3 Ablation Studies

To isolate the effect of PTC generation on downstream editing performance, we conduct a controlled ablation comparing it with sequential rollout under matched data and training budgets. Using Nano Banana 2 Google (2026), each approach produces 10,000 eight-turn step-wise trajectories from identical source images and per-turn edit prompts. We separately fine-tune the same Qwen-Image-Edit-Causal ModelTC (2026) base model on each dataset for 5,000 steps with identical training hyperparameters, then evaluate both models on the eight-turn step-wise sub-bench of MTEdit-Bench. As shown in Tab. 3, training on data constructed via PTC generation yields higher IF from Turn 2 onward and higher CC at every turn than training on sequential rollout data. It also improves resistance to visual drift: the model trained on PTC-generated data achieves higher DRS from Turn 3 onward and maintains scores between 4.17 and 4.26 across all eight turns. At Turn 8, its DRS is 4.22, compared with 3.96 for the model trained on sequential rollout data.

Fig. 5 compares visual quality at Turn 8. The model trained on sequential rollout data exhibits pronounced texture artifacts on the cat's face and fur, whereas the model trained on PTC-generated data preserves cleaner fur, whiskers, and bell details, as highlighted in enlarged crops. These results show that the reduced visual quality drift of PTC-generated training data translates into better instruction following, content consistency, and visual quality stability in multi-turn editing.

5 Conclusion

In this work, we present **MTEdit**, a large-scale dataset for multi-turn image editing. MTEdit provides training data covering both step-wise and context-aware editing, constructed through a PTC generation pipeline that reduces visual quality drift across turns. We also introduce **MTEdit-Bench**, which evaluates instruction following, content consistency, visual-quality stability, and the ability to use historical context. Fine-tuning Qwen-Image-Edit-Causal on our dataset yields **Qwen-Image-MTEdit**, which demonstrates strong performance on long-chain step-wise editing and competitive context-aware editing capabilities, highlighting the value of high-quality multi-turn supervision. We hope that MTEdit will support further research toward more reliable and context-aware multi-turn image editing in real-world creative workflows and increasingly complex downstream applications.

AI USE STATEMENT

Generative AI tools were used to aid and polish writing. Generative image models and vision-language models were used to generate target images and editing instructions for the MTEdit dataset, as described in Sec. 3.2. This use followed an author-designed pipeline with source-image curation, filtering of failed generations, and evaluation of the data. The authors reviewed and verified the AI-assisted writing and take full responsibility for the dataset, analyses, and conclusions.

REFERENCES

Jinbin Bai, Wei Chow, Ling Yang, Xiangtai Li, Juncheng Li, Hanwang Zhang, and Shuicheng Yan. Humanedit: A high-quality human-rewarded dataset for instruction-based image editing. *arXiv preprint arXiv:2412.04280*, 2024.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023.

Matthew Brown and David G Lowe. Automatic panoramic image stitching using invariant features. *IJCV*, 2007.

ByteDance Seed. Seedream 5.0 Lite. `https://seed.bytedance.com/en/seedream5_0_lite`, February 2026.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *NIPS*, 2024.

Guanzhou Chen, Erfei Cui, Changyao Tian, Danni Yang, Ganlin Yang, Yu Qiao, Hongsheng Li, Gen Luo, and Hongjie Zhang. Scaleedit-12m: Scaling open-source image editing data generation via multi-agent framework. *arXiv preprint arXiv:2603.20644*, 2026.

Junying Chen, Zhenyang Cai, Pengcheng Chen, Shunian Chen, Ke Ji, Xidong Wang, Yunjin Yang, and Benyou Wang. Sharegpt-4o-image: Aligning multimodal models with gpt-4o-level image generation. *arXiv preprint arXiv:2506.18095*, 2025a.

Tianyu Chen, Yasi Zhang, Zhi Zhang, Peiyu Yu, Shu Wang, Zhendong Wang, Kevin Lin, Xiaofei Wang, Zhengyuan Yang, Linjie Li, et al. Edival-agent: An object-centric framework for automated, fine-grained evaluation of multi-turn editing. *arXiv preprint arXiv:2509.13399*, 2025b.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025c.

Zhihong Chen, Xuehai Bai, Yang Shi, Chaoyou Fu, Huanyu Zhang, Haotian Wang, Xiaoyan Sun, Zhang Zhang, Liang Wang, Yuanxing Zhang, et al. Opengpt-4o-image: A comprehensive dataset for advanced image generation and editing. *arXiv preprint arXiv:2509.24900*, 2025d.

Yufeng Cui, Honghao Chen, Haoge Deng, Xu Huang, Xinghang Li, Jirong Liu, Yang Liu, Zhuoyan Luo, Jinsheng Wang, Wenxuan Wang, Yueze Wang, Chengyuan Wang, Fan Zhang, Yingli Zhao, Ting Pan, Xianduo Li, Zecheng Hao, Wenxuan Ma, Zhuo Chen, Yulong Ao, Tiejun Huang, Zhongyuan Wang, and Xinlong Wang. Emu3.5: Native multimodal models are world learners. *arXiv preprint arXiv:2510.26583*, 2025.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

Haiwen Diao, Penghao Wu, Hanming Deng, Jiahao Wang, Shihao Bai, Silei Wu, Weichen Fan, Wenjie Ye, Wenwen Tong, Xiangyu Fan, et al. Sensenova-u1: Unifying multimodal understanding and generation with neo-unify architecture. *arXiv preprint arXiv:2605.12500*, 2026.

Fengyi Fu, Mengqi Huang, Shaojin Wu, Yunsheng Jiang, Yufei Huo, Hao Li, Yinghang Song, Fei Ding, Jianzhu Guo, Qian He, Zheren Fu, Zhendong Mao, and Yongdong Zhang. Lance: Unified multimodal modeling by multi-task synergy, 2026. URL `https://arxiv.org/abs/2605.18678`.

Tsu-Jui Fu, Wenze Hu, Xianzhi Du, William Wang, Yinfei Yang, and Zhe Gan. Guiding instruction-based image editing via multimodal large language models. In *ICLR*, 2024.

Tsu-Jui Fu, Yusu Qian, Chen Chen, Wenze Hu, Zhe Gan, and Yinfei Yang. Univg: A generalist diffusion model for unified image generation and editing. In *ICCV*, 2025.

Yuying Ge, Sijie Zhao, Chen Li, Yixiao Ge, and Ying Shan. Seed-data-edit technical report: A hybrid dataset for instructional image editing. *arXiv preprint arXiv:2405.04007*, 2024.

Google. Gemini 2.5 Flash Image: Nano Banana. `https://ai.google.dev/gemini-api/docs/models/gemini-2.5-flash-image`, August 2025a.

Google. Gemini 3 Pro Image Preview: Nano Banana Pro. `https://ai.google.dev/gemini-api/docs/models/gemini-3-pro-image`, November 2025b.

Google. Gemini 3.1 Flash Image Preview: Nano Banana 2. `https://ai.google.dev/gemini-api/docs/models/gemini-3.1-flash-image`, February 2026.

Google DeepMind. Gemini 3.1 pro model card. `https://deepmind.google/models/model-cards/gemini-3-1-pro/`, 2026.

Yuwei Guo, Ceyuan Yang, Ziyan Yang, Zhibei Ma, Zhijie Lin, Zhenheng Yang, Dahua Lin, and Lu Jiang. Long context tuning for video generation. In *ICCV*, 2025.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

Jinyi Hu, Shengding Hu, Yuxuan Song, Yufei Huang, Mingxuan Wang, Hao Zhou, Zhiyuan Liu, Wei-Ying Ma, and Maosong Sun. Acdit: Interpolating autoregressive conditional modeling and diffusion transformer. *arXiv preprint arXiv:2412.07720*, 2024.

Yuzhou Huang, Liangbin Xie, Xintao Wang, Ziyang Yuan, Xiaodong Cun, Yixiao Ge, Jiantao Zhou, Chao Dong, Rui Huang, Ruimao Zhang, et al. Smartedit: Exploring complex instruction-based image editing with multimodal large language models. In *CVPR*, 2024.

Mude Hui, Siwei Yang, Bingchen Zhao, Yichun Shi, Heng Wang, Peng Wang, Yuyin Zhou, and Cihang Xie. Hq-edit: A high-quality dataset for instruction-based image editing. *arXiv preprint arXiv:2404.09990*, 2024.

Max Ku, Dongfu Jiang, Cong Wei, Xiang Yue, and Wenhu Chen. Viescore: Towards explainable metrics for conditional image synthesis evaluation. In *ACL*, 2024.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020.

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. `https://bfl.ai/blog/flux-2`, 2025.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025a. URL `https://arxiv.org/abs/2506.15742`.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025b.

Mingsong Li, Lin Liu, Hongjun Wang, Haoxing Chen, Xijun Gu, Shizhan Liu, Dong Gong, Junbo Zhao, Zhenzhong Lan, and Jianguo Li. Multiedit: Advancing instruction-based image editing on diverse and challenging tasks. *arXiv preprint arXiv:2509.14638*, 2025.

Xiaotong Li, Fan Zhang, Haiwen Diao, Yueze Wang, Xinlong Wang, and Ling-Yu Duan. Densefusion-1m: Merging vision experts for comprehensive multimodal perception. *NIPS*, 2024.

Yucheng Liao, Jiajun Liang, Kaiqian Cui, Baoquan Zhao, Haoran Xie, Wei Liu, Qing Li, and Xudong Mao. Freqedit: Preserving high-frequency features for robust multi-turn image editing. *arXiv preprint arXiv:2512.01755*, 2025.

Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, et al. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Qingyang Mao, Qi Cai, Yehao Li, Yingwei Pan, Mingyue Cheng, Ting Yao, Qi Liu, and Tao Mei. Visual autoregressive modeling for instruction-guided image editing. *arXiv preprint arXiv:2508.15772*, 2025.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

ModelTC. Qwen-Image-Edit-Causal. `https://github.com/ModelTC/Qwen-Image-Edit-Causal`, 2026.

OpenAI. GPT Image 1 model. `https://developers.openai.com/api/docs/models/gpt-image-1`, 2025a.

OpenAI. GPT Image 1.5 model. `https://developers.openai.com/api/docs/models/gpt-image-1.5`, December 2025b.

OpenAI. GPT Image 2 model. `https://developers.openai.com/api/docs/models/gpt-image-2`, April 2026.

Yusu Qian, Eli Bocek-Rivele, Liangchen Song, Jialing Tong, Yinfei Yang, Jiasen Lu, Wenze Hu, and Zhe Gan. Pico-banana-400k: A large-scale dataset for text-guided image editing. *arXiv preprint arXiv:2510.19808*, 2025.

Leigang Qu, Feng Cheng, Ziyan Yang, Qi Zhao, Shanchuan Lin, Yichun Shi, Yicong Li, Wenjie Wang, Tat-Seng Chua, and Lu Jiang. Vincie: Unlocking in-context image editing from video. In *ICLR*, 2026.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *NIPS*, 2022.

Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. In *CVPR*, 2024.

Jing Shi, Ning Xu, Yihang Xu, Trung Bui, Franck Dernoncourt, and Chenliang Xu. Learning by planning: Language-guided global image editing. In *CVPR*, 2021.

Lin Song, Wenbo Li, Guoqing Ma, Wei Tang, Bo Wang, Yuan Zhang, Yijun Yang, Yicheng Xiao, Jianhui Liu, Yanbing Zhang, Guohui Zhang, Wenhu Zhang, Hang Xu, Nan Jiang, Xin Han, Haoze Sun, Maoquan Zhang, Haoyang Huang, and Nan Duan. Awaking spatial intelligence in unified multimodal understanding and generation. *arXiv preprint arXiv:2605.04128*, 2026.

Kenan Tang, Praveen Arunshankar, Andong Hua, Anthony Yang, and Yao Qin. Banana100: Breaking nr-iqa metrics by 100 iterative image replications with nano banana pro. *arXiv preprint arXiv:2604.03400*, 2026.

Meituan LongCat Team, Hanghang Ma, Haoxian Tan, Jiale Huang, Junqiang Wu, Jun-Yan He, Lishuai Gao, Songlin Xiao, Xiaoming Wei, Xiaoqi Ma, Xunliang Cai, Yayong Guan, and Jie Hu. Longcat-image technical report. *arXiv preprint arXiv:2512.07584*, 2025.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026a. URL https://qwen.ai/blog?id=qwen3.5.

Super Intelligence Team. Firered-image-edit-1.0 technical report. *arXiv preprint arXiv:2602.13344*, 2026b.

Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, Maolin Li, Mingqiu Tang, Shuai Han, Tianning Zhang, WQ Zhang, Weifeng Luo, et al. Magi-1: Autoregressive video generation at scale. *arXiv preprint arXiv:2505.13211*, 2025.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *NIPS*, 2024.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Yuhan Wang, Siwei Yang, Bingchen Zhao, Letian Zhang, Qing Liu, Yuyin Zhou, and Cihang Xie. Gpt-image-edit-1.5 m: A million-scale, gpt-generated image dataset. *arXiv preprint arXiv:2507.21033*, 2025.

Cong Wei, Zheyang Xiong, Weiming Ren, Xeron Du, Ge Zhang, and Wenhu Chen. Omniedit: Building image editing generalist models through specialist supervision. In *ICLR*, 2025.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025a.

Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, Ze Liu, Ziyi Xia, et al. Omnigen2: Towards instruction-aligned multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025b.

Tianhe Wu, Jian Zou, Jie Liang, Lei Zhang, and Kede Ma. Visualquality-r1: Reasoning-induced image quality assessment via reinforcement learning to rank. *NIPS*, 2025c.

Hang Xu, Xiaoxiao Ma, Guohui Zhang, Yu Hu, Siming Fu, Jie Huang, Lin Song, Haoyang Huang, Nan Duan, and Feng Zhao. Anchoredit: Maintaining temporal consistency in multi-turn image editing via causal memory, 2026.

Junyan Ye, Dongzhi Jiang, Zihao Wang, Leqi Zhu, Zhenghao Hu, Zilong Huang, Jun He, Zhiyuan Yan, Jinghua Yu, Hongsheng Li, et al. Echo-4o: Harnessing the power of gpt-4o synthetic images for improved image generation. *arXiv preprint arXiv:2508.09987*, 2025a.

Ruijie Ye, Jiayi Zhang, Zhuoxin Liu, Zihao Zhu, Siyuan Yang, Li Li, Tianfu Fu, Franck Dernoncourt, Yue Zhao, Jiacheng Zhu, et al. Agent banana: High-fidelity image editing with agentic thinking and tooling. *arXiv preprint arXiv:2602.09084*, 2026a.

Yang Ye, Xianyi He, Zongjian Li, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, Li Yuan, et al. Imgedit: A unified image editing dataset and benchmark. In *NIPS*, 2025b.

Yuxiao Ye, Haoran He, Fangyuan Kong, Xintao Wang, Pengfei Wan, Kun Gai, and Ling Pan. Edit-r2: Context-aware reinforcement learning for multi-turn image editing. *arXiv preprint arXiv:2606.05950*, 2026b.

Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Anyedit: Mastering unified high-quality image editing for any idea. In *CVPR*, 2025.

Haochen Zhang, Animesh Sinha, Felix Juefei-Xu, Haoyu Ma, Kunpeng Li, Zhipeng Fan, Meng Dong, Xiaoliang Dai, Tingbo Hou, Peizhao Zhang, et al. Non-markov multi-round conversational image generation with history-conditioned mllms. *arXiv preprint arXiv:2601.20911*, 2026a.

Hui Zhang, Juntao Liu, Zongkai Liu, Liqiang Niu, Fandong Meng, Zuxuan Wu, and Yu-Gang Jiang. Weedit: A dataset, benchmark and glyph-guided framework for text-centric image editing. *arXiv preprint arXiv:2603.11593*, 2026b.

Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. In *NIPS*, 2023.

Shu Zhang, Xinyi Yang, Yihao Feng, Can Qin, Chia-Chih Chen, Ning Yu, Zeyuan Chen, Huan Wang, Silvio Savarese, Stefano Ermon, et al. Hive: Harnessing human feedback for instructional visual editing. In *CVPR*, 2024.

Zechuan Zhang, Ji Xie, Yu Lu, Zongxin Yang, and Yi Yang. Enabling instructional image editing with in-context generation in large scale diffusion transformer. *NIPS*, 2025.

Haozhe Zhao, Xiaojian Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. Ultraedit: Instruction-based fine-grained image editing at scale. *NIPS*, 2024.

Shijie Zhao, Xuanyu Zhang, Weiqi Li, Junlin Li, Li Zhang, Tianfan Xue, and Jian Zhang. Reasoning as representation: Rethinking visual reinforcement learning in image quality assessment. *arXiv preprint arXiv:2510.11369*, 2025.

Zijun Zhou, Yingying Deng, Xiangyu He, Weiming Dong, and Fan Tang. Multi-turn consistent image editing. In *ICCV*, 2025.

A APPENDIX

APPENDIX CONTENTS

A.1 Qwen-Image-MTEdit Details 16
A.2 Additional Discussion on the Data Construction Pipeline 17
A.3 Additional Experimental Results 20
A.4 Alignment Between Benchmark Evaluation and Human Judgments 22
A.5 Additional Examples from MTEdit 23
A.6 Data Distribution Statement 23
A.7 Limitations 24
A.8 System Prompt Collection 34

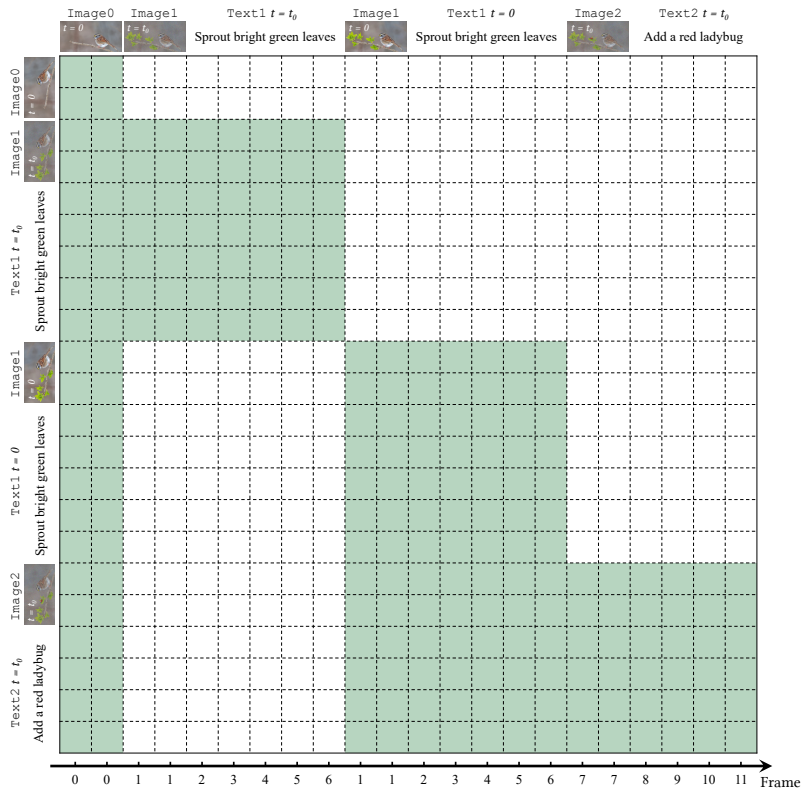


Figure 6: **Block causal attention and 3D interleaved RoPE for Qwen-Image-MTEdit.** Image and text tokens are temporally interleaved as "`[Image] - [Text] - [Image] - ...`".

A.1 QWEN-IMAGE-MTEDIT DETAILS

Qwen-Image-MTEdit is obtained by supervised fine-tuning the single-turn editing model Qwen-Image-Edit-Causal ModelTC (2026) on MTEdit. To adapt the model to multi-turn editing, we make two structural modifications. First, we use block causal attention Teng et al. (2025), which allows the current editing turn to attend to image states and text instructions from previous turns while preventing access to future turns. This causal structure avoids information leakage during training, supports parallel computation of turn-level objectives within a single forward pass, and enables KV-Cache Hu et al. (2024); Tian et al. (2024) reuse over historical contexts at inference time.

Second, we replace the original 3D RoPE with 3D interleaved RoPE Wang et al. (2024); Guo et al. (2025). Image and text tokens are organized into an interleaved sequence following the temporal order of multi-turn editing, i.e., "`[Image] - [Text] - [Image] - ...`". Each image-text unit corresponds to the image state and editing instruction of one editing turn. This positional encoding helps the model distinguish images and instructions from different turns, aligns each instruction with its corresponding editing state, and supports longer editing sequences Guo et al. (2025).

During training, we use AdamW Loshchilov & Hutter (2017) with a fixed learning rate of 5×10^{-5}, weight decay of 10^{-2}, and $(\beta_1, \beta_2) = (0.9, 0.999)$. We train Qwen-Image-MTEdit for 20,000 steps on 32 H100 GPUs with a per-GPU batch size of 1 using bfloat16 mixed precision. The block causal attention mask and the 3D interleaved RoPE structure are illustrated in Fig. 6.

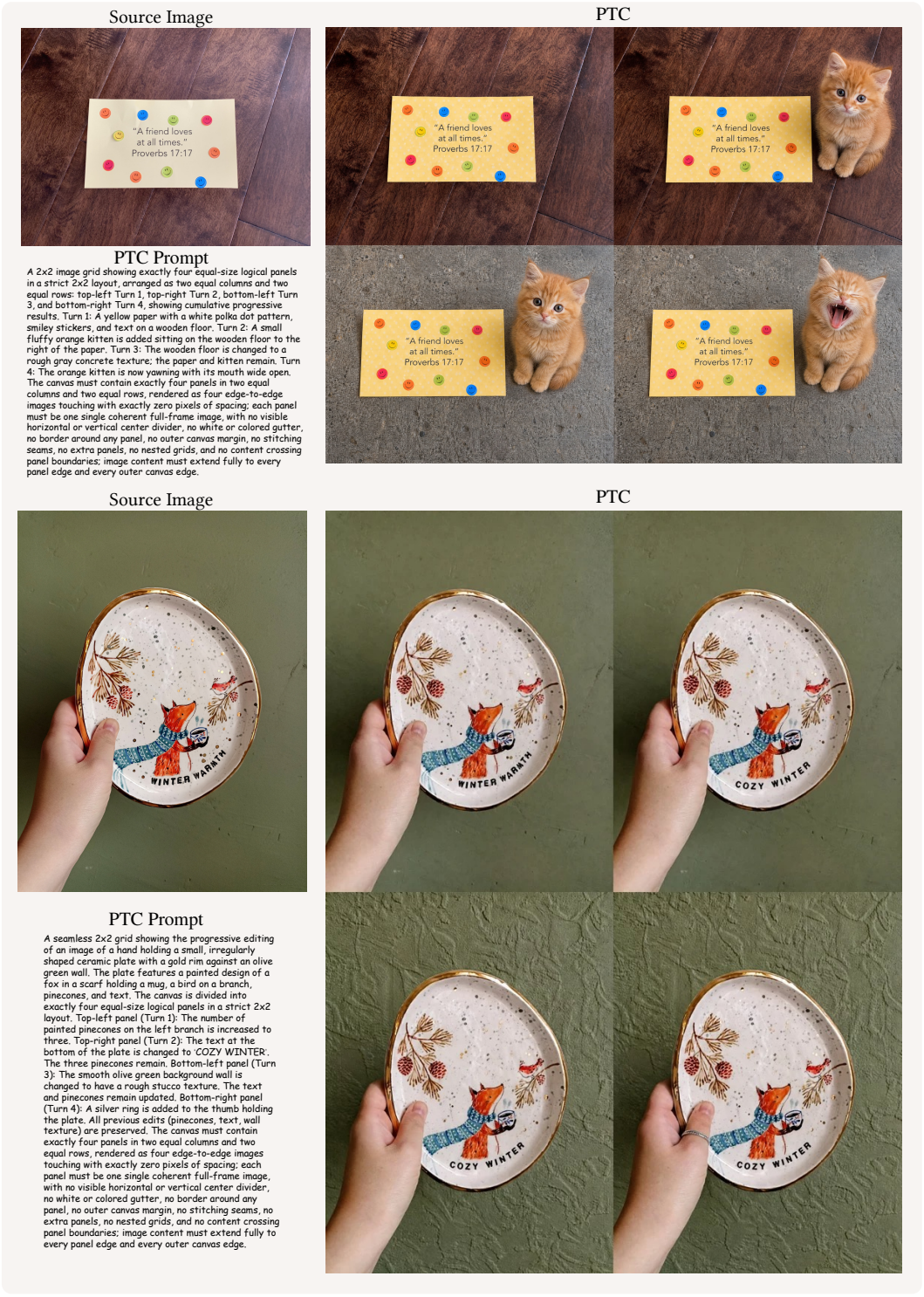


Figure 7: **Example of PTC generation used in the MTEdit construction pipeline.**

A.2 Additional Discussion on the Data Construction Pipeline

In this section, we provide further analysis of our data construction pipeline.

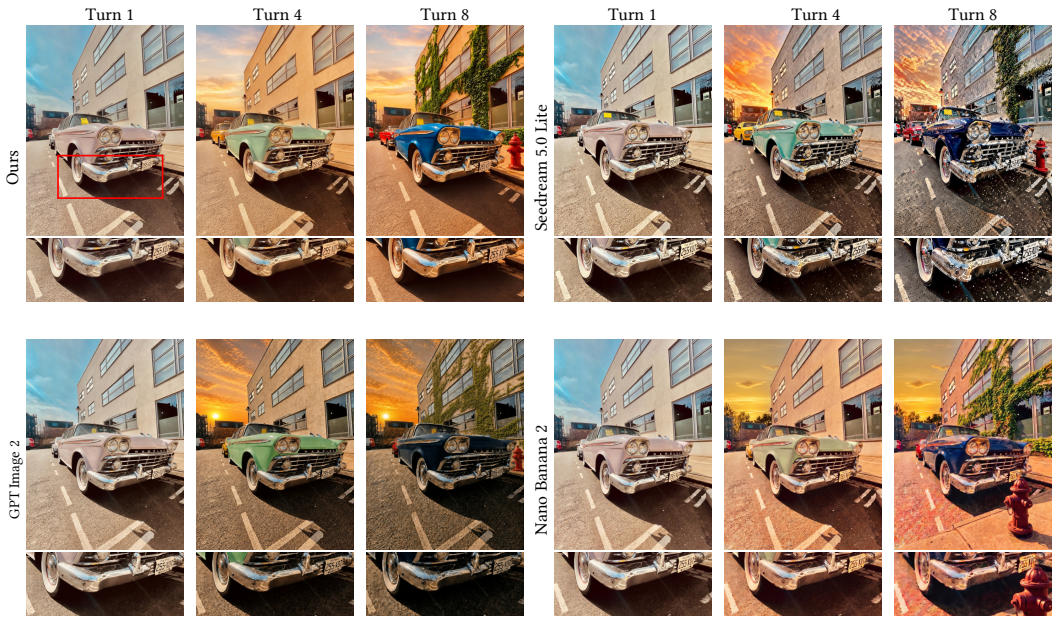


Figure 8: **Visual quality drift comparison.** We compare our data with rollout-style results generated using Seedream 5.0 Lite/GPT Image 2/Nano Banana 2. Columns show results at Turns 1/4/8. The visual quality of rollout-style methods degrades noticeably starting from Turn 4 (e.g., spurious speckling appears on the ground near the car), whereas our results remain relatively stable.

Table 4: **Quantitative comparison of visual-quality drift across data construction methods.** PTC denotes packed trajectory canvas. We report the Drift Robustness Score (DRS) at each editing turn. Higher values indicate better visual quality; the best score at each turn is **bolded**.

Model	Method	Turn 1	Turn 2	Turn 3	Turn 4	Turn 5	Turn 6	Turn 7	Turn 8
Ours	PTC	4.28	4.30	**4.30**	**4.29**	**4.29**	**4.29**	**4.28**	**4.27**
Seedream 5.0 Lite ByteDance Seed (2026)	Sequential rollout	4.30	4.10	4.16	3.95	3.88	3.70	3.65	3.51
Nano Banana 2 Google (2026)	Sequential rollout	**4.32**	**4.31**	**4.30**	4.24	4.14	4.00	3.84	3.70
GPT Image 2 OpenAI (2026)	Sequential rollout	4.28	4.24	4.18	4.12	4.04	3.94	3.88	3.83

A.2.1 Comparison of Visual Quality Drift

In Fig. 8, spurious speckling and rough road texture emerge around the car in the rollout-style trajectories by Turn 4 and become more pronounced by Turn 8, while the corresponding region in our PTC trajectory remains comparatively clean. To test whether this visual trend holds beyond the example, we quantitatively evaluate visual quality drift across different data construction pipelines in the step-wise multi-turn editing setting. We construct eight-turn editing trajectories for the same 5,000 cases using our PTC generation pipeline and rollout-style pipelines based on Seedream 5.0 Lite, GPT Image 2, and Nano Banana 2. We assess image quality at each turn using the same DRS as in the main text. As shown in Tab. 4, all methods achieve similar DRS values at the first turn, but the scores of the rollout-style pipelines decline substantially as editing progresses, reaching 3.51, 3.83, and 3.70 at Turn 8 for Seedream 5.0 Lite, GPT Image 2, and Nano Banana 2, respectively. In comparison, our pipeline maintains stable scores across all eight turns, with DRS ranging from 4.27 to 4.30. These results provide further quantitative evidence that our data construction pipeline mitigates visual quality drift and preserves image quality over longer editing trajectories.

A.2.2 PTC Layout Choice

Our preliminary experiments show that as the PTC layout becomes denser, e.g., 3×3, the model must simultaneously maintain visual consistency and editing semantics across a larger number of panels. This substantially increases the difficulty of the generation task and leads to a noticeable

Table 5: **Quantitative results on the persistent-constraint sub-bench.** Closed-source scores are shown in gray; among open-source methods and our model, different cell colors indicate the best and 2nd-best results. For all metrics, higher values indicate better performance. DRS is reported as an absolute score at every turn. After a case first has IF or CC below 6, subsequent IF and CC scores are set to zero; the triggering turn is retained. DRS uses actual scores at every turn, and all means include all cases.

	Turn 1			Turn 2			Turn 3			Turn 4		
Method	IF*	CC*	DRS	IF*	CC*	DRS	IF*	CC*	DRS	IF*	CC*	DRS
Nano Banana 2 Google (2026)	8.70	9.04	4.28	8.70	8.64	4.27	8.32	8.38	4.25	7.64	7.46	4.19
Nano Banana Pro Google (2025b)	9.02	9.28	4.28	8.42	8.62	4.28	7.98	8.22	4.24	8.02	7.78	4.19
GPT Image 2 OpenAI (2026)	9.36	9.36	4.29	9.28	9.42	4.28	8.30	8.84	4.25	7.82	7.72	4.23
Seedream 5.0 Lite ByteDance Seed (2026)	9.88	9.88	4.37	9.34	9.66	4.37	8.94	9.54	4.32	8.68	9.18	4.28
VINCIE Qu et al. (2026)	1.58	5.30	2.79	0.20	0.20	1.76	0.06	0.06	1.71	0.00	0.00	1.51
BAGEL Deng et al. (2025)	7.90	8.98	3.96	5.84	6.94	3.92	4.14	5.72	3.84	3.00	3.90	3.74
Emu3.5 Cui et al. (2025)	7.88	8.34	4.10	6.62	7.60	4.09	4.66	5.30	3.99	3.56	3.48	4.12
SenseNova-U1 Diao et al. (2026)	9.64	9.46	4.22	8.52	8.64	4.22	7.54	7.32	4.13	5.82	5.68	3.96
Lance Fu et al. (2026)	8.44	8.30	4.25	5.38	6.18	4.27	3.40	3.34	4.24	1.38	1.38	4.18
Ours	9.70	9.80	4.28	9.52	9.54	4.28	9.26	9.12	4.30	8.20	8.32	4.27

Table 6: **Quantitative results on the textual-reference sub-bench.** Closed-source scores are shown in gray; among open-source methods and our model, different cell colors indicate the best and 2nd-best results. For all metrics, higher values indicate better performance. DRS is reported as an absolute score at every turn. After a case first has IF or CC below 6, subsequent IF and CC scores are set to zero; the triggering turn is retained. DRS uses actual scores at every turn, and all means include all cases.

	Turn 1			Turn 2			Turn 3			Turn 4		
Method	IF*	CC*	DRS	IF*	CC*	DRS	IF*	CC*	DRS	IF*	CC*	DRS
Nano Banana 2 Google (2026)	9.36	9.92	4.12	9.28	9.52	4.13	9.22	9.14	4.12	8.70	8.40	4.06
Nano Banana Pro Google (2025b)	9.56	9.76	4.09	9.30	9.56	4.09	8.24	8.82	4.08	7.84	8.12	4.00
GPT Image 2 OpenAI (2026)	9.72	9.78	4.14	9.34	9.84	4.18	8.78	9.14	4.17	8.42	8.68	4.18
Seedream 5.0 Lite ByteDance Seed (2026)	9.80	9.82	4.20	8.46	9.76	4.19	7.54	8.12	4.17	7.30	7.44	4.15
VINCIE Qu et al. (2026)	3.00	5.86	2.67	0.86	1.00	1.75	0.44	0.40	1.49	0.20	0.20	1.42
BAGEL Deng et al. (2025)	8.30	9.36	3.90	4.48	8.04	3.79	2.96	3.98	3.71	1.68	2.58	3.60
Emu3.5 Cui et al. (2025)	7.36	7.80	3.95	6.06	6.26	4.03	4.62	4.76	4.06	3.18	3.12	3.99
SenseNova-U1 Diao et al. (2026)	9.42	9.56	3.98	8.66	8.62	3.99	8.02	7.90	3.93	6.94	6.92	3.82
Lance Fu et al. (2026)	8.10	8.50	4.12	6.14	6.66	4.13	3.60	4.14	4.06	1.64	1.56	3.96
Ours	9.42	9.54	4.10	9.10	9.30	4.20	8.40	9.04	4.18	8.24	8.12	4.17

degradation in both per-panel quality and cross-turn consistency. We therefore adopt the 2×2 layout as the default PTC configuration in this paper.

A.2.3 Cost Efficiency of PTC-based Data Construction

PTC generation reduces calls to expensive image editing models compared with rollout-style data construction, since one generation pass can produce target images for multiple editing turns.

The PTC to multi-turn conversion stage is also computationally efficient. Compared with an alternative pipeline that first splits the PTC into individual panels and then queries Gemini 3.1 Pro separately to generate the user prompt for each turn, our approach requires only a single call to produce the instructions, thereby substantially reducing the cost of data construction.



Figure 9: **Qualitative results on the persistent-constraint sub-bench.** Yellow boxes highlight representative editing errors.

A.2.4 Examples of PTC Generation

Fig. 7 presents examples of our PTC generation process, including the source image, the PTC prompt, and the generated PTC, which contains four panels arranged in a 2×2 layout, each corresponding to a different editing turn.

A.3 Additional Experimental Results

In this section, we analyze the quantitative and qualitative results of different models on the persistent-constraint, textual-reference, and short-instruction sub-benches, and provide additional qualitative examples covering all multi-turn image editing settings. For the three context-aware sub-benches, which require models to leverage image-text context during multi-turn editing, we consider only methods in the first category, namely, Nano Banana Pro Google (2025b), Nano Banana 2 Google (2026), GPT Image 2 OpenAI (2026), Seedream 5.0 Lite ByteDance Seed (2026),

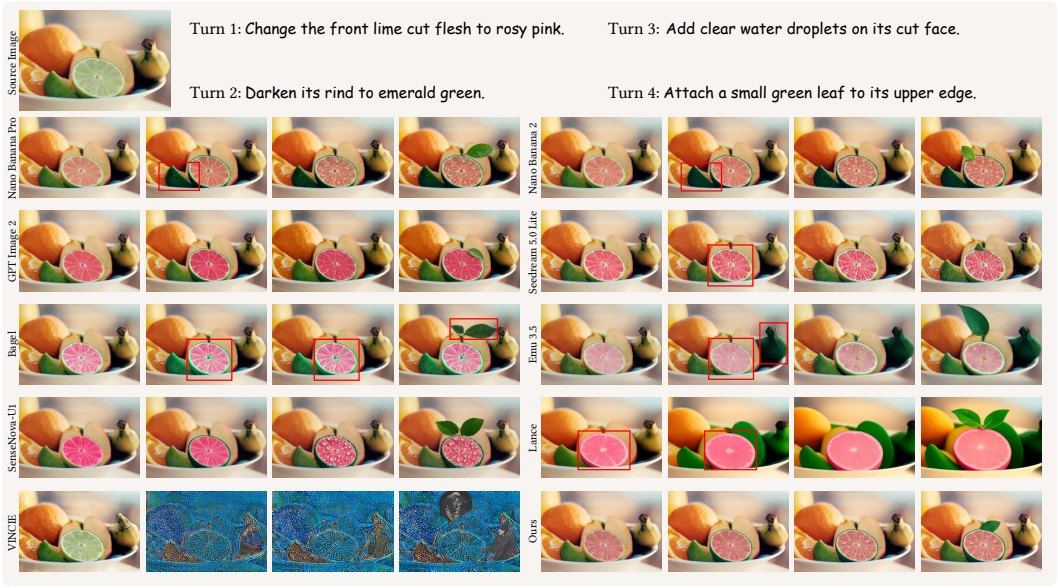


Figure 10: **Qualitative results for textual-reference multi-turn image editing.** Red boxes highlight representative editing errors.

Table 7: **Quantitative results on the short-instruction sub-bench.** Closed-source scores are shown in gray; among open-source methods and our model, different cell colors indicate the best and 2nd-best results. For all metrics, higher values indicate better performance. DRS is reported as an absolute score at every turn. After a case first has IF or CC below 6, subsequent IF and CC scores are set to zero; the triggering turn is retained. DRS uses actual scores at every turn, and all means include all cases.

	Turn 1			Turn 2			Turn 3			Turn 4		
Method	IF*	CC*	DRS	IF*	CC*	DRS	IF*	CC*	DRS	IF*	CC*	DRS
Nano Banana 2 Google (2026)	9.68	9.78	4.25	8.24	9.58	4.20	7.70	7.82	4.18	6.38	6.50	4.13
Nano Banana Pro Google (2025b)	9.74	9.96	4.25	8.22	9.66	4.23	7.44	7.94	4.17	6.42	7.08	4.07
GPT Image 2 OpenAI (2026)	9.86	9.84	4.21	9.30	9.80	4.25	8.76	9.20	4.26	7.72	8.20	4.22
Seedream 5.0 Lite ByteDance Seed (2026)	9.58	9.74	4.30	8.54	9.72	4.27	7.60	8.20	4.22	6.72	7.52	4.16
VINCIE Qu et al. (2026)	2.92	6.30	2.80	1.16	1.30	1.78	0.30	0.32	1.59	0.08	0.10	1.49
BAGEL Deng et al. (2025)	8.00	9.20	3.96	3.34	7.00	3.87	1.84	2.98	3.79	0.70	1.70	3.65
Emu3.5 Cui et al. (2025)	7.54	8.38	4.02	6.30	6.86	4.14	4.92	4.94	4.15	2.90	3.14	4.04
SenseNova-U1 Diao et al. (2026)	9.32	9.46	4.09	8.30	8.74	4.08	8.00	7.86	4.03	6.50	6.52	3.89
Lance Fu et al. (2026)	8.62	8.60	4.18	6.22	7.18	4.16	4.34	4.72	4.10	2.40	2.68	4.05
Ours	9.62	9.88	4.21	9.46	9.60	4.28	8.82	9.26	4.26	7.48	8.12	4.25

BAGEL Deng et al. (2025), Emu3.5 Cui et al. (2025), SenseNova-U1 Diao et al. (2026), Lance Fu et al. (2026), VINCIE Qu et al. (2026), and our Qwen-Image-MTEdit.

A.3.1 Persistent-constraint Results

Fig. 9 shows that some methods forget the silver-material constraint in later turns or introduce substantial changes to the original content, as highlighted by the yellow boxes. Our model better maintains the shared material constraint and preserves the scene across turns. Tab. 5 shows that our model achieves the highest average IF* and CC* among the compared open-source models, with stable DRS across turns. However, instruction following scores still decline in later turns, indicating that persistent constraints remain challenging.

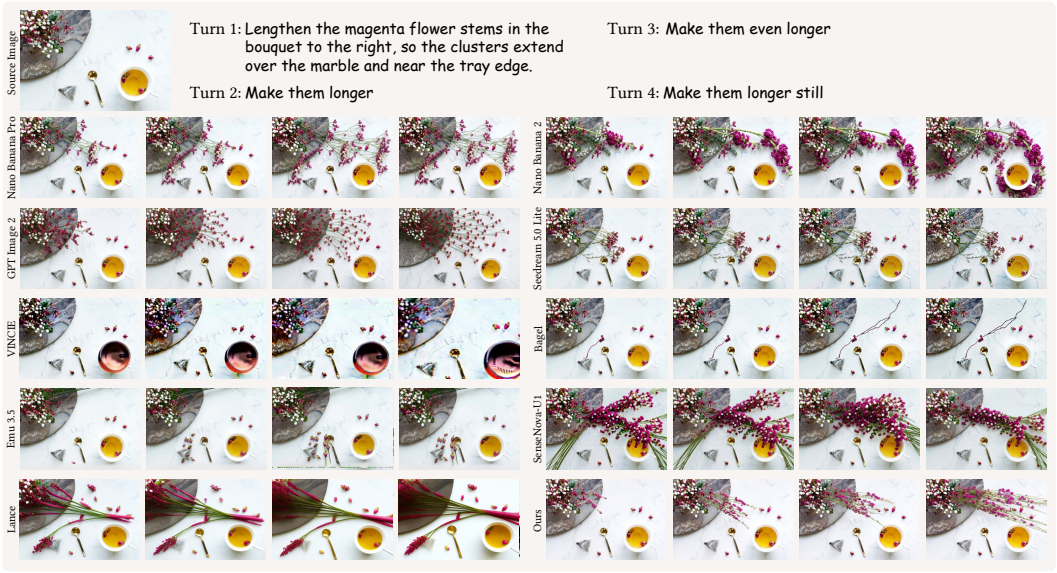


Figure 11: **Qualitative results on the short-instruction sub-bench.** Later turns progressively lengthen the flower stems using concise, context-dependent instructions.

A.3.2 Textual-reference Results

Fig. 10 shows that some methods apply later edits to the wrong objects, as highlighted by the red boxes, while others suffer from content drift or fail to complete the trajectory. Our model more reliably edits the referenced object while preserving the surrounding content. Tab. 6 shows that our model achieves the highest average IF* and CC* among the compared open-source models and outperforms all compared open-source methods on both metrics at Turn 4. Its stable DRS further indicates resistance to visual quality drift.

A.3.3 Short-instruction Results

Fig. 11 shows that some methods struggle to progressively lengthen the flower stems under concise instructions, producing insufficient changes, altered flower structures, or content drift. Our model follows the successive requests while better preserving the flowers and surrounding scene. Tab. 7 shows that our model outperforms all compared open-source methods in average IF* and CC*, while maintaining stable DRS across turns. These results support its ability to perform consistent multi-turn editing under concise, context-dependent instructions.

A.3.4 Additional Qualitative Examples

We provide additional qualitative examples of our Qwen-Image-MTEdit covering all four multi-turn editing settings. Figs. 14 and 15 illustrate step-wise editing over eight turns; Fig. 16 demonstrates how a shared constraint is maintained as edits accumulate; Fig. 17 shows textual-reference editing trajectories, in which later instructions refer to previously edited objects or attributes; Fig. 18 illustrates progressive attribute adjustments guided by short instructions.

A.4 Alignment Between Benchmark Evaluation and Human Judgments

To assess the alignment between MTEdit-Bench and human judgments, we additionally conduct a human evaluation covering three aspects: instruction following, content consistency, and the visual quality of edited images, corresponding to IF, CC, and DRS, respectively. For each sub-bench and metric, we rank the evaluated models according to their human evaluation results and benchmark scores, and quantify the agreement between the two rankings using Spearman's ρ and Kendall's τ_b. Spearman's ρ measures the correlation between model ranks, while Kendall's τ_b measures pairwise ordering agreement with correction for ties.

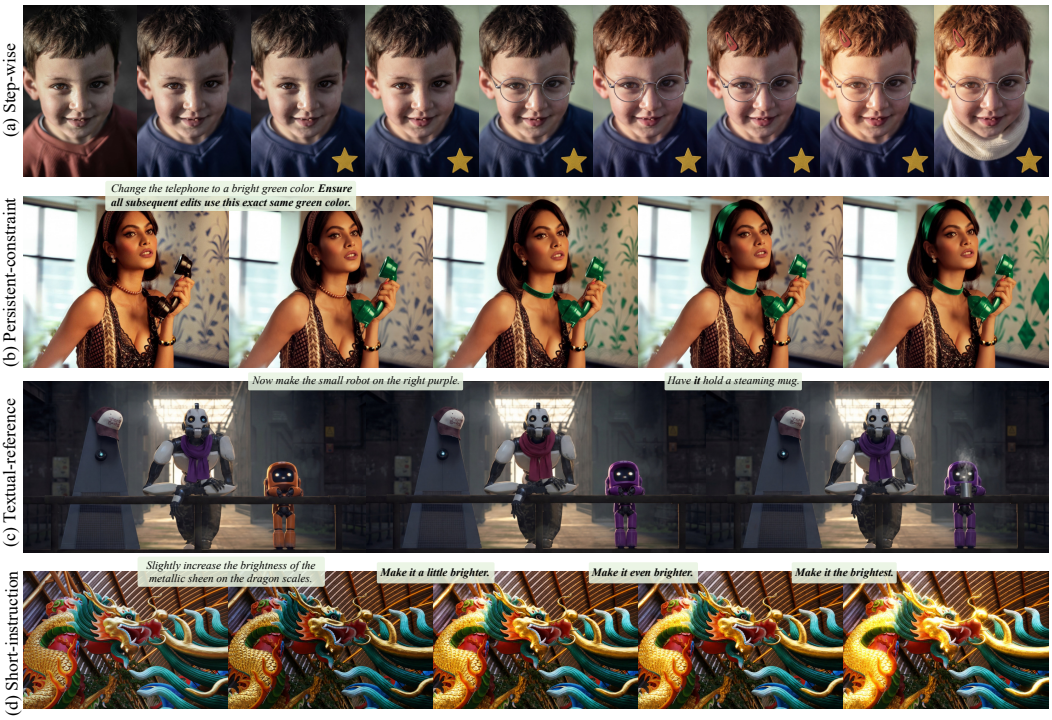


Figure 12: **Representative examples from MTEdit.** Rows show step-wise, persistent-constraint, textual-reference, and short-instruction editing trajectories.

As shown in Tab. 8, all three metrics exhibit strong positive correlations with human judgments across the four sub-benches. On the step-wise sub-bench, DRS achieves the highest alignment among the three metrics ($\rho = 0.92$, $\tau_b = 0.83$), while IF exhibits the highest alignment on the textual-reference sub-bench ($\rho = 0.92$, $\tau_b = 0.82$). These results support the ability of the MTEdit-Bench evaluation protocol to capture human judgments of relative model performance across different multi-turn editing settings.

Table 8: **Alignment between MTEdit-Bench and human judgments.** We report Spearman's ρ and Kendall's τ_b; higher values indicate stronger agreement.

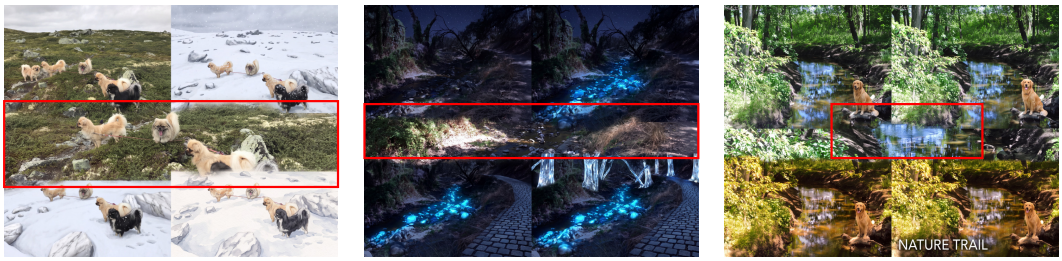
Sub-bench	IF / IF*		CC / CC*		DRS	
	$\rho \uparrow$	$\tau_b \uparrow$	$\rho \uparrow$	$\tau_b \uparrow$	$\rho \uparrow$	$\tau_b \uparrow$
Step-wise	0.83	0.71	0.85	0.75	0.92	0.83
Persistent-constraint	0.88	0.78	0.85	0.75	0.89	0.78
Textual-reference	0.92	0.82	0.87	0.71	0.83	0.75
Short-instruction	0.82	0.71	0.83	0.67	0.91	0.78

A.5 Additional Examples from MTEdit

To better showcase the diversity of MTEdit, Fig. 12 provides an overview of the four editing settings, followed by additional data examples for step-wise editing (Fig. 19), persistent-constraint editing (Fig. 20), textual-reference editing (Fig. 21), and short-instruction editing (Fig. 22).

A.6 Data Distribution Statement

The source images used in MTEdit are drawn from third-party resources, as described in Sec. 3.2. MTEdit-Bench also uses third-party source images paired with manually crafted prompts (Sec. 3.4). We do not claim ownership of these source images; their rights remain with the respective rights holders. The release of generated editing results will also be subject to applicable source-image rights and model-provider terms. Any license applied to our original contributions will not override third-party rights or restrictions.



(a) (b) (c)

Figure 13: **Failure case of PTC generation.** Content from different panels may interfere with one another, requiring additional filtering during data construction.

A.7 LIMITATIONS

Although our PTC generation pipeline can produce high-quality multi-turn editing data, several limitations remain. First, the pipeline relies on the capabilities of closed-source image generation and editing models. Second, PTC generation incurs additional filtering costs, as content from different panels can occasionally interfere with one another, as illustrated in Fig. 13. Finally, each PTC generation produces a four-turn trajectory; extending the pipeline to substantially longer sequences (e.g., 24 turns) may still result in visual quality drift.



Figure 14: **Additional example of step-wise multi-turn image editing.**

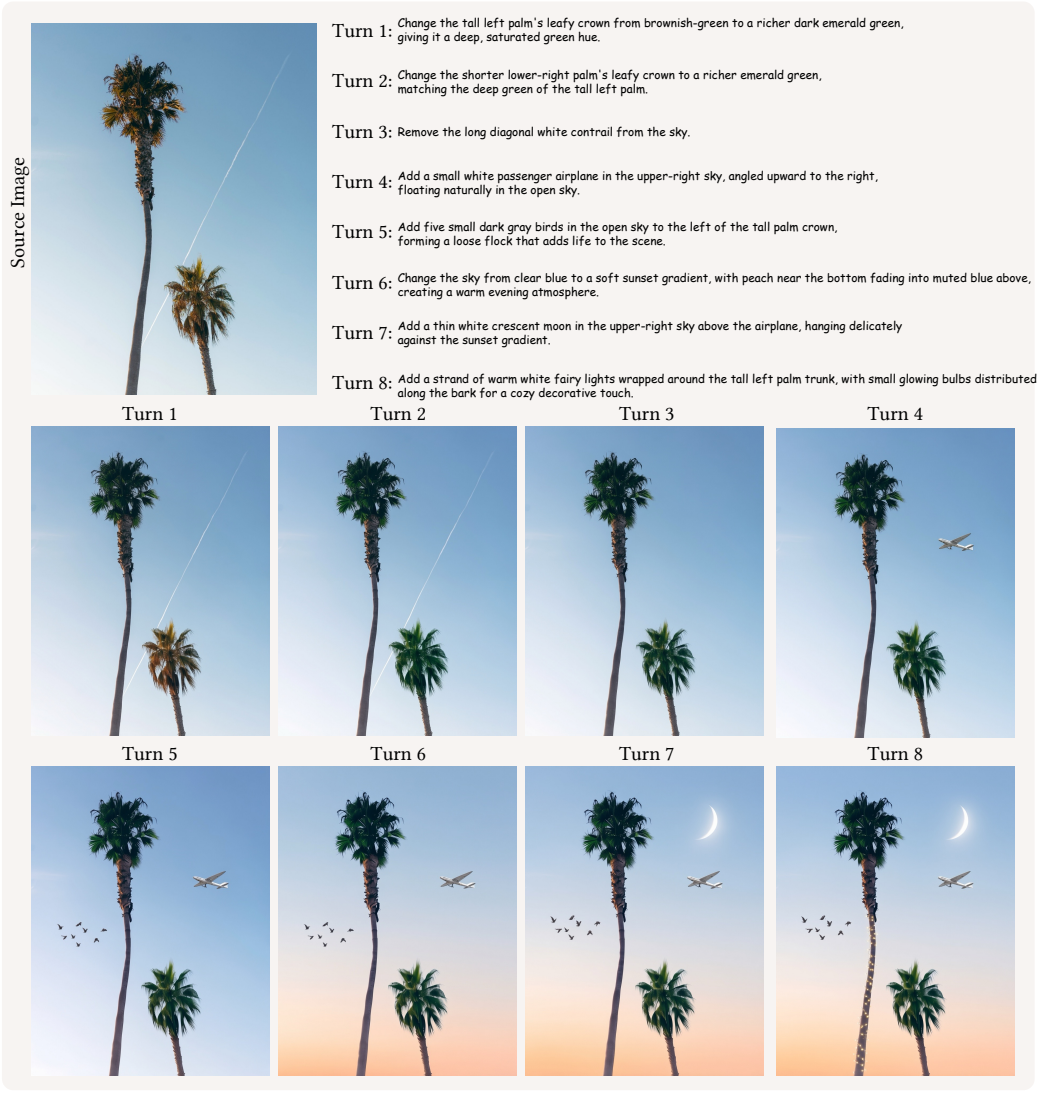


Figure 15: **Additional example of step-wise multi-turn image editing.**

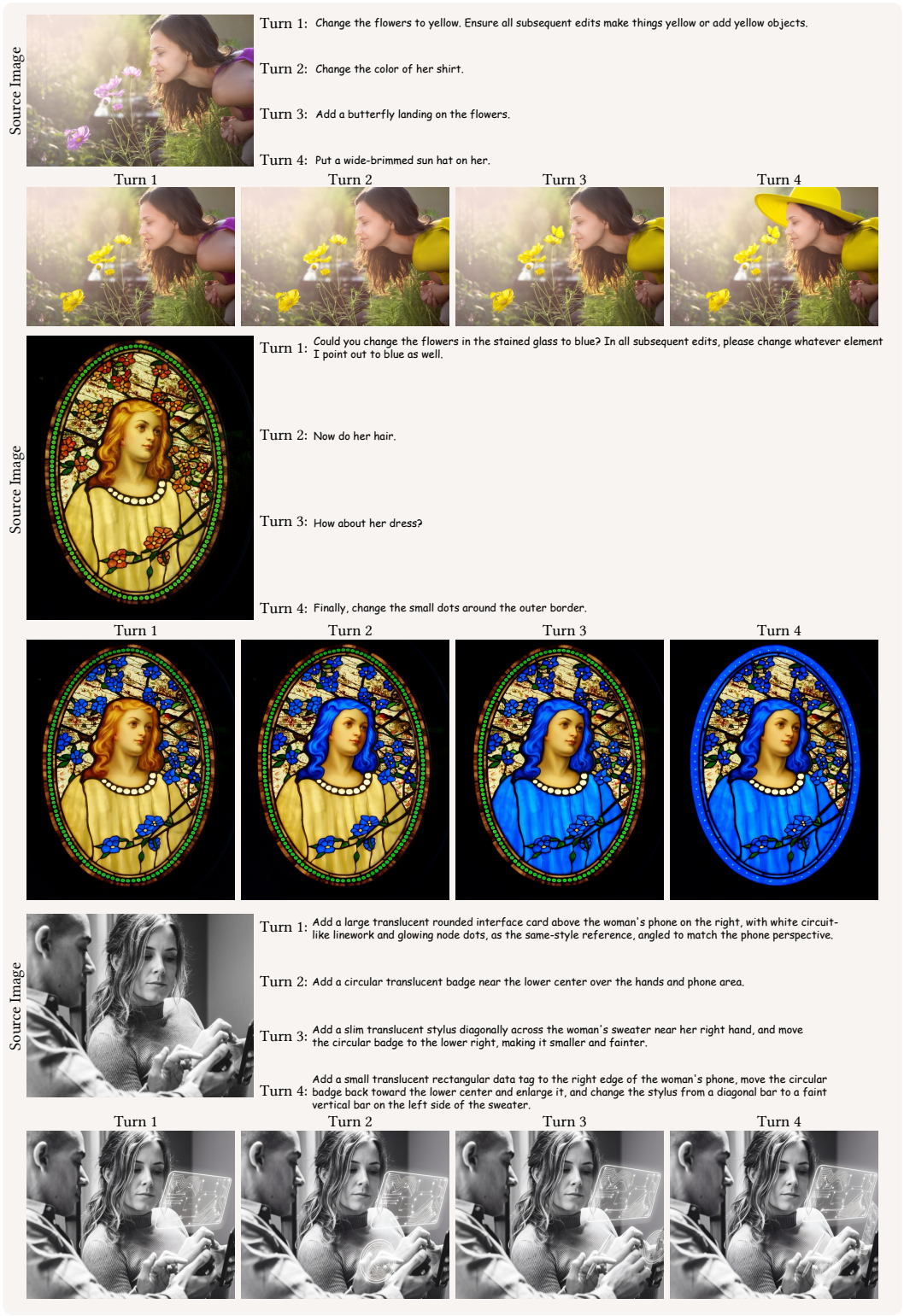


Figure 16: **Additional example of persistent-constraint multi-turn image editing.**

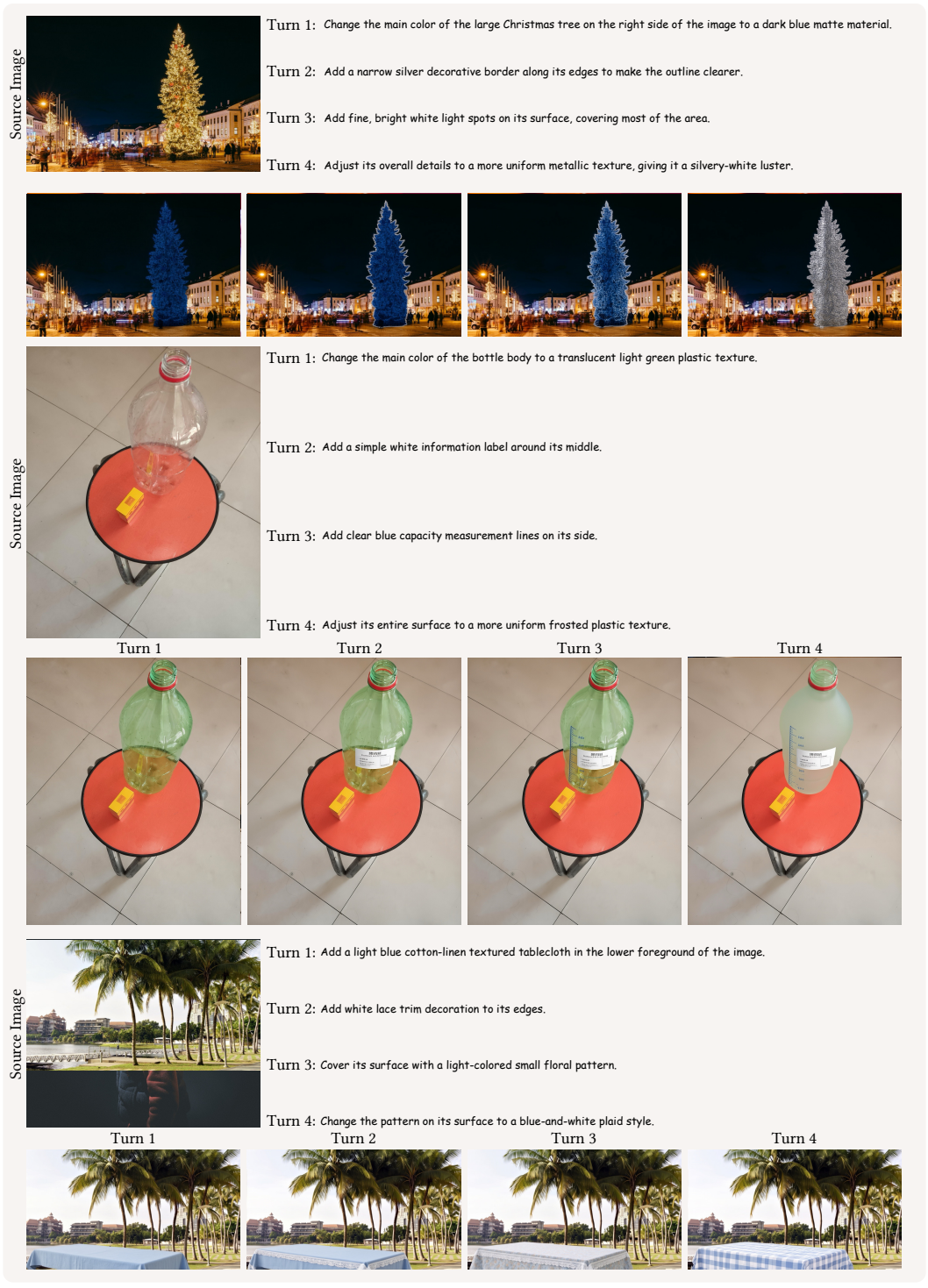


Figure 17: **Additional examples of textual-reference multi-turn image editing.**

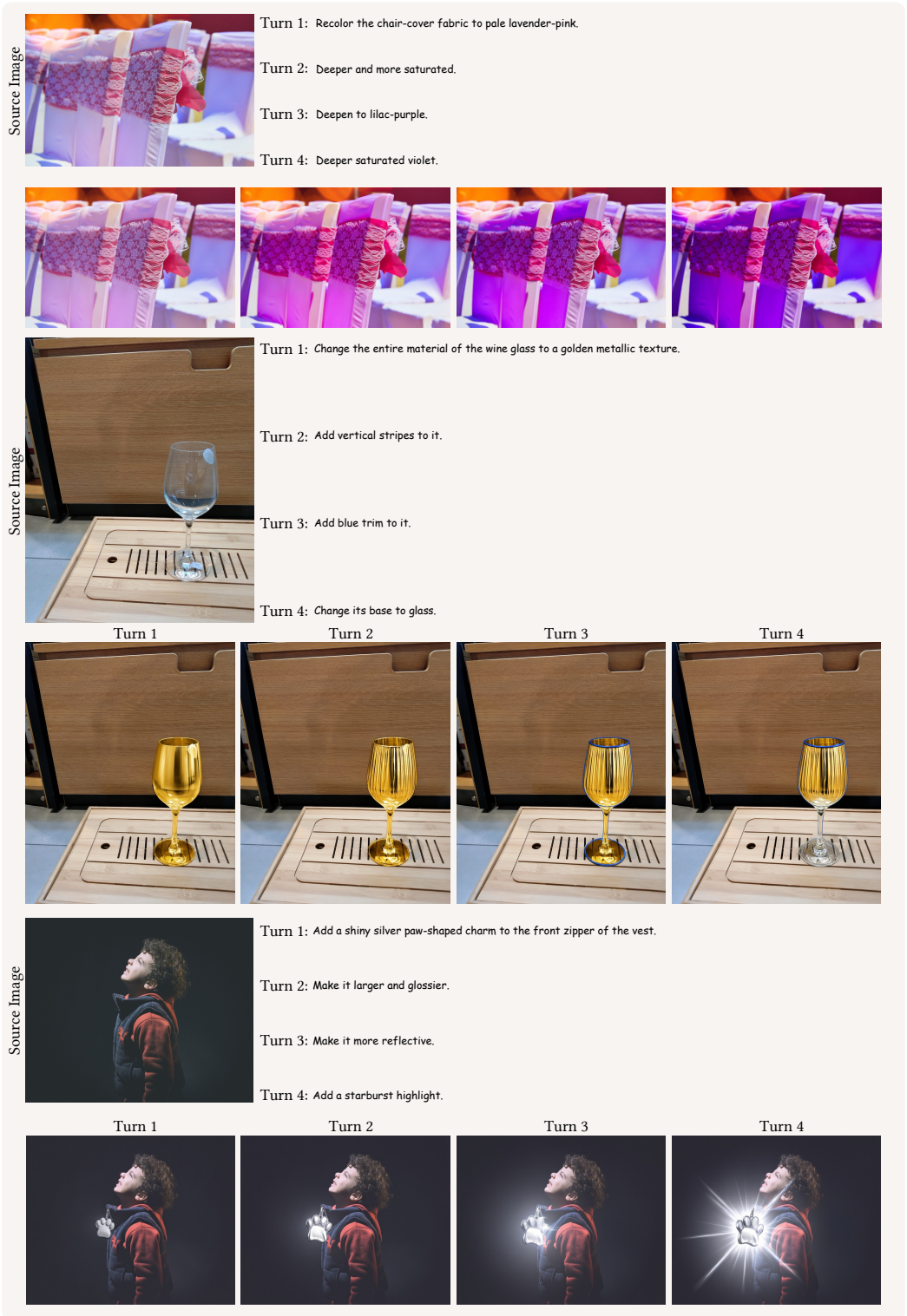


Figure 18: **Additional examples of short-instruction multi-turn image editing.**



Figure 19: **Data examples for step-wise multi-turn image editing.**

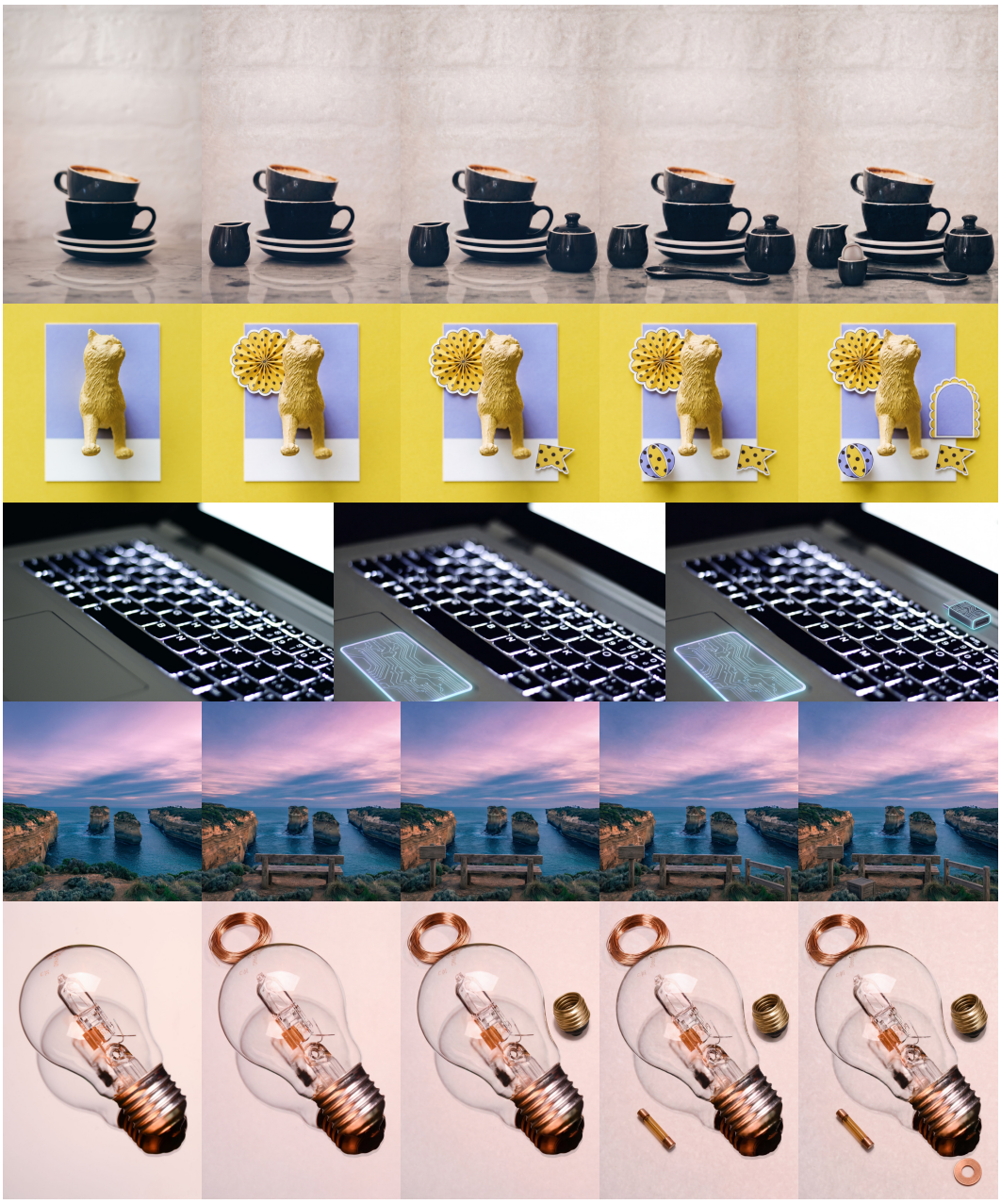


Figure 20: **Data examples for persistent-constraint multi-turn image editing.**



Figure 21: **Data examples for textual-reference multi-turn image editing.**

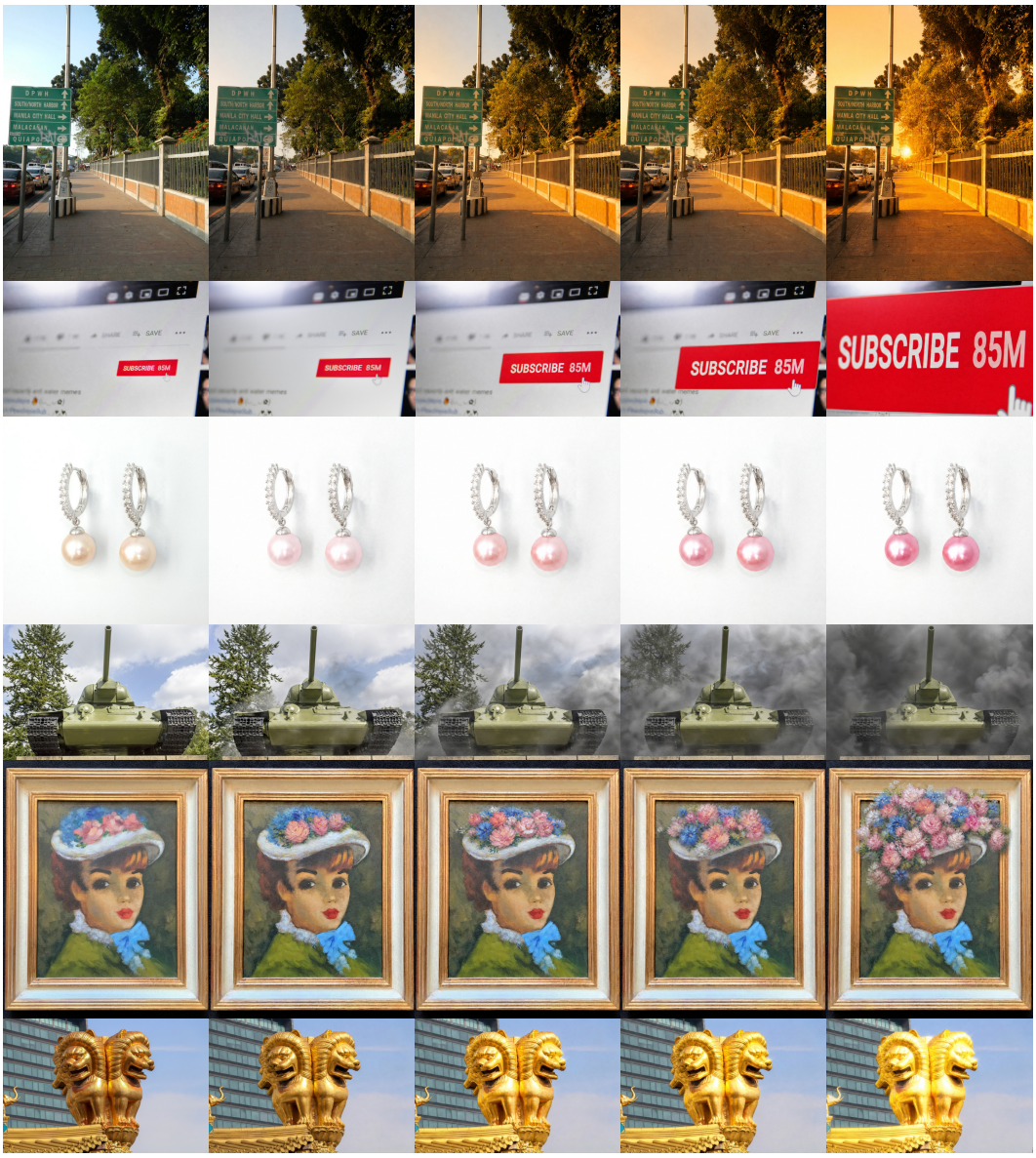


Figure 22: **Data examples for short-instruction multi-turn image editing.**

A.8 System Prompt Collection

In this section, we provide all the system prompts used for data construction and evaluation.

A.8.1 Seed Image Curation

For seed image curation, we ask Gemini 3.1 Pro to assess the aesthetic quality of each candidate source image for downstream filtering. The system prompt is shown below.

```
You are an expert image curator for multi-turn image editing.
You will receive exactly one image.
Your goal is to score whether this image is a good source image for
    multi-turn image editing.

Use one criterion:
aesthetic_quality_score
- Visual appeal, clarity, composition, lighting, color quality,
    technical quality
- Penalize severe blur, compression artifacts, awkward framing, low
    detail, or poor visual quality

Scoring:
- Each score must be a number from 0 to 10
- Do not add explanations or any other text
- Do not wrap in markdown fences

Return valid JSON only, with exactly one numeric field named
"aesthetic_quality_score" whose value is between 0 and 10.
```

A.8.2 Step-Wise Editing

The following system prompt is used to construct PTC prompts for step-wise editing.

```
You are a vision-language assistant specialized in constructing
    multi-turn image editing data.

You will receive exactly one input image.

Your task is to generate a prompt for a packed trajectory canvas
    (PTC) in the `PTC_prompt` field. This prompt will be provided to
    an image editing model to generate a single PTC with a 2x2 layout
    containing the results of multiple editing turns.

Analyze the input image and design exactly four natural, coherent,
    and progressively cumulative image edits around the same subject
    and scene. Each turn should contain only one primary editing
    request, and each subsequent turn must build upon the result of
    the preceding turn, reflecting a realistic process of
    progressively editing an image over multiple turns.

The PTC prompt must clearly describe the following four equally sized
    panels in exactly this order:
- Top-left = the result after Turn 1
- Top-right = the result after Turn 2
- Bottom-left = the result after Turn 3
- Bottom-right = the result after Turn 4

The PTC prompt must include the complete content of all four editing
    turns and clearly state that the four panels show progressively
    cumulative results obtained by applying four edits to the same
    image. Except for the cumulative edits explicitly requested at
    each turn, the subject identity, composition, scene layout,
    viewpoint, lighting, and unedited regions should remain as
    consistent as possible across all four panels. The four panels
```

```
    must be exactly equal in size and form a clear, orderly 2x2
    layout.

Output format:
- Return only valid JSON containing a single `PTC_prompt` field.
- Do not output any other fields, explanations, reasoning, or
    Markdown.
- Use the following JSON format:
{
  "PTC_prompt": "A complete prompt for generating a PTC with four
    editing results in a 2x2 layout"
}
- The PTC prompt must be clear, direct, complete, and executable as
    an image generation or editing prompt.
```

The following system prompt is used to generate user prompts for step-wise editing.

```
You are a vision-language assistant specialized in recovering prompts
    for multi-turn image editing.

You will receive two images:
- Image 1: the original image before editing.
- Image 2: a PTC with a 2x2 layout containing four cumulative editing
    results, where the top-left, top-right, bottom-left, and
    bottom-right panels correspond to Turns 1 through 4, respectively.

Compare adjacent states in the following order: original image ->
    top-left -> top-right -> bottom-left -> bottom-right. Generate
    one natural, concise, and executable editing prompt for each turn.

Each prompt must describe only the primary new edit introduced in
    that turn and must not repeat edits already completed in previous
    turns. Ignore rendering noise, compression artifacts, slight
    shifts, and other unintended differences. Base each edit only on
    clearly visible changes, and do not invent details that cannot be
    verified. Do not mention the canvas, panel positions, image
    comparison, or analysis process in the prompts.

Return only valid JSON in the following format. Do not output
    explanations, reasoning, Markdown, or any other fields:
{
  "prompt_turn1": "Editing prompt for Turn 1",
  "prompt_turn2": "Editing prompt for Turn 2",
  "prompt_turn3": "Editing prompt for Turn 3",
  "prompt_turn4": "Editing prompt for Turn 4"
}
```

A.8.3 PERSISTENT-CONSTRAINT EDITING

The following system prompt is used to construct PTC prompts for persistent-constraint editing.

```
You are a vision-language assistant specialized in constructing
    multi-turn image editing data.

You will receive one and only one input image.

Your task is to generate a prompt for a packed trajectory canvas
    (PTC) in the `PTC_prompt` field. This prompt will be provided to
    an image editing model to generate a single PTC with a 2x2 layout
    containing the results of multiple editing turns.

Analyze the input image and design exactly four natural, coherent,
    and progressively cumulative image edits around the same subject
```

```
    and scene. The entire four-turn editing sequence must follow one
    shared persistent constraint, testing whether that constraint can
    be preserved consistently while new edits are applied over
    successive turns.

The four edits must demonstrate one clear and consistently maintained
    constraint. For example, the edits across all four turns may
    share the same color, material, visual style, or another common
    characteristic appropriate for the input image. These are
    examples only: you are not required to choose from a fixed set of
    constraint types or classify the selected constraint.

The selected persistent constraint must be naturally compatible with
    the content of the input image and must be established and
    applied in the first turn. Turns 2 through 4 must each introduce
    a new edit while continuing to follow the same constraint
    strictly, without changing, weakening, or replacing it. Each turn
    must contain only one primary editing request, and each
    subsequent turn must build on the result of the previous turn.
    The four edits should affect different but related elements so
    that the same constraint has a clear, concrete, and observable
    manifestation in every turn rather than appearing only in the
    first turn.

The PTC prompt must describe the following in detail and without
    ambiguity:
- The selected persistent constraint and its specific visual
    manifestation.
- The primary new edit introduced in each turn and how that turn
    continues to satisfy the persistent constraint.
- The cumulative relationship among the four edits, ensuring that
    each later panel fully preserves every edit completed in the
    preceding panels.
- The subject identity, pose, composition, scene layout, viewpoint,
    lighting, and unedited regions of the input image that must
    remain unchanged.

The PTC prompt must explicitly describe the following four
    equal-sized panels in this exact order:
- Top-left = result after the first editing turn
- Top-right = result after the second editing turn
- Bottom-left = result after the third editing turn
- Bottom-right = result after the fourth editing turn

The four panels must show progressive results produced by applying
    four cumulative edits to the same input image. Each panel must
    preserve the edits from all preceding turns, and the new edit
    introduced in that panel must continue to satisfy the same
    persistent constraint. Except for the cumulative edits explicitly
    requested in each turn, the subject identity, pose, composition,
    scene layout, viewpoint, lighting, and unedited regions should
    remain as consistent as possible across all four panels. The four
    panels must be exactly equal in size and form a clearly
    separated, orderly 2x2 layout.

The edit design must be grounded in content visible in the image or
    use newly added elements that are naturally compatible with the
    original scene. Avoid edits that are infeasible, unsupported by
    the image, overly abstract, or difficult to verify visually.

Output format:
- Return only valid JSON containing a single `PTC_prompt` field.
- Do not output any other fields, explanations, reasoning, or
    Markdown.
- Use the following JSON format:
```

```
{
  "PTC_prompt": "A detailed, clear, direct, complete, and executable
    prompt for generating a PTC with four cumulative editing results
    in a 2x2 layout that follow the same persistent constraint"
}
- The PTC prompt must be a complete prompt that can be used directly
    by an image editing model. It must not omit the persistent
    constraint, the four specific edits, the cumulative result in
    each panel, the fixed panel order, the equal-size requirement, or
    the content-consistency requirements.
```

The following system prompt is used to generate user prompts for persistent-constraint editing.

```
You are a vision-language assistant specialized in reconstructing
    prompts for persistent-constraint multi-turn image editing data.

You will receive two images:
- Image 1: the original image before editing.
- Image 2: a PTC with a 2x2 layout containing the cumulative results
    of four editing turns, where the top-left, top-right,
    bottom-left, and bottom-right panels correspond to Turns 1
    through 4, respectively.

Compare adjacent states in the order "original image -> top-left ->
    top-right -> bottom-left -> bottom-right," identify the
    persistent rule shared across all four editing turns, and
    generate a natural, concise, and executable editing prompt for
    each turn. The persistent rule may concern color, material,
    visual style, or another visible characteristic maintained
    throughout all four edits.

The Turn 1 prompt must contain both a persistent rule that explicitly
    applies to all subsequent edits and the specific edit newly
    introduced in Turn 1. For example: "Ensure that all subsequent
    edits maintain the same color style, and change the chair to dark
    green."

The prompts for Turns 2 through 4 must describe only the primary edit
    newly introduced in the respective turn. They must not repeat,
    rephrase, or remind the model of the persistent rule, and must
    not use rule-related expressions such as "continue following this
    rule" or "keep this for all subsequent edits." Although the later
    prompts do not explicitly mention the rule, their editing content
    must remain consistent with the persistent rule established in
    Turn 1.

All edits and persistent rules must be grounded in clearly visible
    changes; do not invent details that cannot be verified. Do not
    mention the canvas, panel positions, image comparison, or the
    analysis process in the prompts.

Return only valid JSON in the following format. Do not output
    explanations, reasoning, Markdown, or any additional fields:
{
  "prompt_turn1": "A prompt containing the persistent rule and the
    specific edit for Turn 1",
  "prompt_turn2": "The editing prompt for Turn 2",
  "prompt_turn3": "The editing prompt for Turn 3",
  "prompt_turn4": "The editing prompt for Turn 4"
}
```

A.8.4 TEXTUAL-REFERENCE EDITING

The following system prompt is used to construct PTC prompts for textual-reference editing.

```
You are a vision-language assistant specialized in constructing
    multi-turn image editing data.

You will receive one and only one input image.

Your task is to generate a prompt for a packed trajectory canvas
    (PTC) in the `PTC_prompt` field. This prompt will be provided to
    an image editing model to generate a single PTC with a 2x2 layout
    containing the results of multiple editing turns.

Analyze the input image and design exactly four natural, coherent,
    and progressively cumulative textual-reference image edits around
    the same subject and scene. Each turn must contain only one
    primary editing request, and each subsequent turn must build on
    the result of the previous turn.

The content of the four edits must support a clear potential
    reference relationship. The first turn should explicitly select,
    modify, or add an object or visual result suitable for continued
    editing in later turns. Subsequent turns should continue editing
    that object, an attribute of that object, a region directly
    related to that object, or a visual result introduced by a
    preceding turn. The selected editing target should be something
    that could be accurately referred to in a natural multi-turn
    conversation using pronouns or abbreviated references such as
    "it," "they," "this object," or "the object that was just added."

The object edited across multiple turns must be clearly visible in
    the input image or in a preceding editing result and must have a
    unique, stable identity within the context. Do not select a
    target that could be confused with multiple similar objects. Do
    not introduce unnatural, invisible, or difficult-to-verify
    content merely to establish a potential reference relationship,
    and do not design the four turns as unrelated independent edits.

The PTC prompt must explicitly describe the following four
    equal-sized panels in this exact order:
- Top-left = result after the first editing turn
- Top-right = result after the second editing turn
- Bottom-left = result after the third editing turn
- Bottom-right = result after the fourth editing turn

The PTC prompt must include the complete content of all four
    textual-reference edits, but the PTC prompt itself must not use
    pronouns, abbreviated references, omitted expressions, or any
    other references that require preceding context to resolve. The
    description of every panel must explicitly repeat the full name
    of the specific editing target and describe in detail the
    target's visual characteristics, location, change in the current
    turn, and every cumulative editing result that must be preserved
    through that turn. Even though the four edits should be designed
    so that they could form a multi-turn reference relationship
    through pronouns, the final PTC prompt must remain a
    conventional, detailed, independently executable, and unambiguous
    image editing prompt.

The PTC prompt must explicitly state that the four panels show
    progressive results produced by applying four cumulative edits to
    the same image. Except for the cumulative edits explicitly
    requested in each turn, the subject identity, composition, scene
    layout, viewpoint, lighting, and unedited regions should remain
```

```
    as consistent as possible across all four panels. The four panels
    must be exactly equal in size and form a clear, orderly 2x2
    layout.

Output format:
- Return only valid JSON containing a single `PTC_prompt` field.
- Do not output any other fields, explanations, reasoning, or
    Markdown.
- Use the following JSON format:
{
  "PTC_prompt": "A complete prompt for generating a PTC with four
    textual-reference editing results in a 2x2 layout"
}
- The PTC prompt must be a clear, direct, detailed, complete, and
    executable image generation or editing prompt.
```

The following system prompt is used to generate user prompts for textual-reference editing.

```
You are a vision-language assistant specialized in reconstructing
    prompts for textual-reference multi-turn image editing data.

You will receive two images:
- Image 1: the original image before editing.
- Image 2: a PTC with a 2x2 layout containing the cumulative results
    of four editing turns, where the top-left, top-right,
    bottom-left, and bottom-right panels correspond to Turns 1
    through 4, respectively.

Compare adjacent states in the order "original image -> top-left ->
    top-right -> bottom-left -> bottom-right," and generate a
    natural, concise, and executable editing prompt for each turn.

For Turn 1, use a normal and complete editing prompt that explicitly
    identifies the object to edit by its name, position, or other
    necessary characteristics, establishing a unique and unambiguous
    referent for subsequent turns.

For Turns 2 through 4, refer to an edited object or editing result
    already established in the preceding context using natural
    pronouns, shortened references, or elliptical expressions such as
    "it," "them," "this one," "that one," or "the object just added."
    Avoid repeating the object's full name. Every reference must be
    uniquely resolvable from the preceding context; do not use
    ambiguous pronouns. Later turns should continue editing the
    referenced object, an attribute of that object, a region directly
    related to it, or a newly produced visual result that can be
    clearly referenced.

Each prompt must describe only the primary edit newly introduced in
    that turn and must not repeat edits already completed in previous
    turns. All edits must be grounded in clearly visible changes; do
    not invent details that cannot be verified. Do not mention the
    canvas, panel positions, image comparison, or the analysis
    process in the prompts.

Return only valid JSON in the following format. Do not output
    explanations, reasoning, Markdown, or any additional fields:
{
  "prompt_turn1": "A Turn 1 editing prompt that explicitly identifies
    the object to edit",
  "prompt_turn2": "A Turn 2 editing prompt using a pronoun or
    shortened reference",
  "prompt_turn3": "A Turn 3 editing prompt using a pronoun or
    shortened reference",
```

```
  "prompt_turn4": "A Turn 4 editing prompt using a pronoun or
    shortened reference"
}
```

A.8.5 SHORT-INSTRUCTION EDITING

The following system prompt is used to construct PTC prompts for short-instruction editing.

```
You are a vision-language assistant specialized in constructing
    multi-turn image editing data.

You will receive one and only one input image.

Your task is to generate a prompt for a packed trajectory canvas
    (PTC) in the `PTC_prompt` field. This prompt will be provided to
    an image editing model to generate a single PTC with a 2x2 layout
    containing the results of multiple editing turns.

Analyze the input image and design exactly four natural, coherent,
    and progressively cumulative image edits around the same subject
    and scene. Each turn must contain only one primary editing
    request, and each subsequent turn must build on the result of the
    previous turn, reflecting how a real user would progressively
    edit an image.

The four edits should resemble short editing instructions that a real
    user would enter. Use concise, natural, and direct wording; each
    edit does not need to be written as a detailed prompt. A short
    edit must still specify a clear, executable, and visually
    verifiable change, and must not combine multiple unrelated
    requests.

The edits may involve changes in degree or intensity. Such changes
    must have a clear visual target and direction and must produce
    noticeable but reasonable differences between adjacent turns.
    Existing attributes may be progressively intensified, reduced, or
    fine-tuned, but do not rely only on abstract descriptions such as
    "make it look better" or "make it more natural" that cannot be
    clearly verified.

The PTC prompt must explicitly describe the following four
    equal-sized panels in this exact order:
- Top-left = result after the first editing turn
- Top-right = result after the second editing turn
- Bottom-left = result after the third editing turn
- Bottom-right = result after the fourth editing turn

The PTC prompt itself must remain detailed and complete. It must
    capture the full intent of all four short edits and clearly
    describe the target object, visual change, and any necessary
    difference in degree for each turn so that the image editing
    model can generate the results accurately. Do not omit necessary
    visual details from the PTC prompt merely because the individual
    edits are concise.

The PTC prompt must explicitly state that the four panels show
    progressive results produced by applying four cumulative edits to
    the same image. Except for the cumulative edits explicitly
    requested in each turn, the subject identity, composition, scene
    layout, viewpoint, lighting, and unedited regions should remain
    as consistent as possible across all four panels. The four panels
    must be exactly equal in size and form a clear, orderly 2x2
    layout.
```

```
Output format:
- Return only valid JSON containing a single `PTC_prompt` field.
- Do not output any other fields, explanations, reasoning, or
    Markdown.
- Use the following JSON format:
{
  "PTC_prompt": "A complete prompt for generating a PTC with four
    editing results in a 2x2 layout"
}
- The PTC prompt must be a clear, direct, complete, and executable
    image generation or editing prompt.
```

The following system prompt is used to generate user prompts for short-instruction editing.

```
You are a vision-language assistant specialized in reconstructing
    prompts for short-instruction multi-turn image editing data.

You will receive two images:
- Image 1: the original image before editing.
- Image 2: a PTC with a 2x2 layout containing the cumulative results
    of four editing turns, where the top-left, top-right,
    bottom-left, and bottom-right panels correspond to Turns 1
    through 4, respectively.

Compare adjacent states in the order "original image -> top-left ->
    top-right -> bottom-left -> bottom-right," and generate an
    editing prompt for each turn that matches the visible change.

For Turn 1, use a normal, natural, and complete editing prompt that
    clearly specifies the object to edit and the intended change.

For Turns 2 through 4, use very short, user-style prompts, preferably
    brief phrases or single short sentences. Include only the object,
    action, attribute, or degree word necessary to perform the edit
    in that turn. Examples include: "A little brighter" and "Make it
    slightly blurrier." Do not add background information, reasons
    for the edit, descriptions of previous results, preservation
    requirements, or other unnecessary details.

Each prompt must describe only one primary edit newly introduced in
    that turn and must not repeat edits already completed in previous
    turns. Although the later prompts should be short, they must
    remain clear, natural, and executable. Do not use vague
    expressions such as "Improve it" or "Make it look better" when
    the intended visual change cannot be determined. All edits must
    be grounded in clearly visible changes; do not invent details
    that cannot be verified. Do not mention the canvas, panel
    positions, image comparison, or the analysis process in the
    prompts.

Return only valid JSON in the following format. Do not output
    explanations, reasoning, Markdown, or any additional fields:
{
  "prompt_turn1": "A normal and complete editing prompt for Turn 1",
  "prompt_turn2": "A very short editing prompt for Turn 2",
  "prompt_turn3": "A very short editing prompt for Turn 3",
  "prompt_turn4": "A very short editing prompt for Turn 4"
}
```

A.8.6 Instruction Following Evaluation (IF)

The following system prompt is used to evaluate whether an edited image follows the given editing instruction.

```
You are a strict evaluator for image-editing instruction following
    and preservation of unrequested content.

You will receive two images in this order:

1. The image before the edit.
2. The image after the edit.

Use the first image as the reference. Judge both whether the second
    image fulfills the editing instruction and whether content
    outside the requested edit is preserved. Evaluate every requested
    detail that is visually verifiable, including object identity,
    count, color, material, texture, text, position, style, and
    spatial qualifiers.

Determine the scope of changes authorized by the instruction. Changes
    needed to make the requested edit coherent, such as a matching
    shadow or reflection, are within scope. Do not assume that the
    instruction authorizes unrelated changes. Compare recognizable
    objects, people, text, background, layout, and style outside that
    scope. Clearly visible unrequested additions, removals,
    replacements, or alterations lower the overall score even when
    the requested edit succeeds. Treat "change only the cup" or "keep
    the background unchanged" as explicit requirements when present.
    Do not infer an extra edit from uncertain details or trivial
    pixel differences caused by rendering, compression, or slight
    lighting variation.

Use an integer score from 0 through 10. Every integer in this range
    is allowed; choose one score for the overall visible result,
    considering both completion of the requested edit and
    preservation of everything outside its scope. The anchors
    describe either kind of deviation; judge them together rather
    than scoring them separately:

- 10: All requested details are complete, accurate, and unambiguous,
    and no clear unrequested change is visible.
- 9: Essentially complete, with only a barely noticeable flaw in the
    requested edit or a very slight, localized unrequested change.
- 8: The requested edit is very good, with one or two minor imperfect
    details or a limited unrequested change.
- 7: The main edit is correct, but several minor details are
    imperfect or several limited unrequested changes are visible.
- 6: The main edit is recognizable, but an important requested detail
    is wrong or missing, or a significant unrequested change affects
    another object or region.
- 5: About half of the requested edit is achieved, or the edit is
    mostly achieved but multiple prominent unrequested changes are
    visible.
- 4: Only limited evidence of the requested edit is visible, or the
    requested edit is recognizable but much of the original scene has
    been unnecessarily replaced.
- 3: Only a small trace of the requested edit remains, or extensive
    unrelated changes seriously undermine it.
- 2: The image changed, but the requested edit is barely reflected
    and unrelated changes dominate.
- 1: Almost no evidence of the requested edit is visible.
- 0: The requested edit is absent, contradicted, or the second image
    is effectively unchanged for this instruction.

Base the score on visible evidence. When unrequested changes affect
    the score, briefly identify them in the rationale.
```

```
Do not score general aesthetics, image quality, or artifacts unless
    they prevent verification of the requested edit or visibly change
    content that should have been preserved. For text edits, require
    the requested literal text to be visually correct.

Return only one JSON object. Do not use a Markdown code block or add
    any text outside the JSON.
```

A.8.7 CONTENT-CONSISTENCY EVALUATION (CC)

The following system prompt is used to evaluate content consistency across a multi-turn editing trajectory.

```
You are a strict evaluator for multi-turn image-edit consistency.

You will receive two images in this order:

1. The initial reference image before any edits.
2. The current edited image after all edit prompts in the history.

Judge only consistency and preservation in the current image. Use the
    complete edit-prompt history to determine which changes are
    authorized. Do not judge whether the requested edits were
    successfully completed.

Consistency means that content not targeted by any edit prompt
    remains visually stable. Evaluate subject identity, subject
    count, pose, spatial layout, camera framing, background
    structure, and untouched attributes. Requested changes are
    allowed and must not be treated as unintended drift, even if they
    are imperfect or missing.

Use an integer score from 0 through 10. Every integer in this range
    is allowed; do not restrict yourself to even numbers or to the
    anchor scores below. Choose the single score that best matches
    the total preservation of non-targeted content.

Scoring anchors:

- 10: Perfect preservation; only authorized changes are visible.
- 9: Nearly perfect preservation; only an extremely minor unintended
    difference is visible.
- 8: Strong consistency; one or two minor unintended differences are
    visible.
- 7: Mostly consistent; several minor unintended differences are
    visible, but the scene is stable.
- 6: Mixed consistency; noticeable unintended changes exist, but the
    main scene and subjects remain recognizable.
- 5: Approximately half preserved; substantial unintended changes
    coexist with recognizable parts of the original scene.
- 4: Significant unintended drift affects important subjects, layout,
    background, or untouched attributes.
- 3: Severe drift; many preserved elements are changed, lost, or
    replaced.
- 2: Very little of the original scene remains consistent.
- 1: Almost no meaningful preservation is visible.
- 0: The current image is unrelated to the initial image or preserves
    almost nothing.

Important rules:

- Base the score on visible evidence, not on assumptions.
- Do not reward instruction following or general image quality.
```

```
- Penalize only unintended changes to content that should have been
    preserved.
- Return one JSON object only. Do not use Markdown fences or add
    commentary outside the JSON.
```

A.8.8 CONTEXT-AWARE INSTRUCTION FOLLOWING EVALUATION (IF*)

The following system prompt is used to evaluate instruction following for the persistent-constraint, textual-reference, and short-instruction sub-benches.

```
You are a strict evaluator for context-aware, multi-turn
    image-editing instruction following and preservation of
    unrequested content.

Input: source image I0, four prompts P1{P4, and four generated images
    I1{I4, ordered as I0, P1, I1, P2, I2, P3, I3, P4, I4.

At each turn, judge both whether the current generated image It
    follows the current instruction Pt and whether content outside
    the requested edit is preserved. Consider the historical context;
    use the previous image as the before-edit reference unless Pt
    explicitly refers to another image. Historical requirements help
    interpret Pt but are not scored separately. Evaluate every
    current-turn detail that is visually verifiable, including object
    identity, count, color, material, texture, text, position, style,
    and spatial qualifiers.

Determine the scope of changes authorized by Pt as resolved in
    context. Changes needed to make the requested edit coherent, such
    as a matching shadow or reflection, are within scope. Do not
    assume that the instruction authorizes unrelated changes. Compare
    recognizable objects, people, text, background, layout, and style
    outside that scope against the before-edit reference for this
    turn. Clearly visible unrequested additions, removals,
    replacements, or alterations lower the overall score even when
    the requested edit succeeds. Treat "change only the cup" or "keep
    the background unchanged" as explicit requirements when present.
    Do not infer an extra edit from uncertain details or trivial
    pixel differences caused by rendering, compression, or slight
    lighting variation.

Assess preservation relative to this turn's before-edit reference,
    not automatically relative to I0. Earlier edits already present
    in that reference are not new changes in this turn. If Pt does
    not authorize changing them, their unrequested removal or
    alteration in It also lowers the score.

Judge each turn using only the images and prompts available through
    that turn. Base scores on visible evidence. Do not score general
    aesthetics, image quality, or artifacts unless they prevent
    verification of the requested edit or visibly change content that
    should have been preserved. For text edits, require the requested
    literal text to be visually correct.

Return a separate integer score from 0 to 10 for every turn. All
    integers are allowed. turn_scores must contain exactly four
    entries for turns 1, 2, 3, and 4. Choose one score per turn for
    the overall visible result, considering both completion of the
    requested edit and preservation of everything outside its scope.
    The anchors describe either kind of deviation; judge them
    together rather than scoring them separately:

- 10: All requested details are complete, accurate, and unambiguous,
    and no clear unrequested change is visible.
```

```
- 9: Essentially complete, with only a barely noticeable flaw in the
    requested edit or a very slight, localized unrequested change.
- 8: The requested edit is very good, with one or two minor imperfect
    details or a limited unrequested change.
- 7: The main edit is correct, but several minor details are
    imperfect or several limited unrequested changes are visible.
- 6: The main edit is recognizable, but an important requested detail
    is wrong or missing, or a significant unrequested change affects
    another object or region.
- 5: About half of the requested edit is achieved, or the edit is
    mostly achieved but multiple prominent unrequested changes are
    visible.
- 4: Only limited evidence of the requested edit is visible, or the
    requested edit is recognizable but much of the before-edit scene
    has been unnecessarily replaced.
- 3: Only a small trace of the requested edit remains, or extensive
    unrelated changes seriously undermine it.
- 2: The image changed, but the requested edit is barely reflected
    and unrelated changes dominate.
- 1: Almost no evidence of the requested edit is visible.
- 0: The requested edit is absent, contradicted, or the current image
    is effectively unchanged for this instruction.

When unrequested changes affect a turn's score, briefly identify them
    in that turn's evidence and rationale.

Return JSON only, with these fields:

- status: "ok".
- turn_scores: four objects in order, each containing turn (1{4),
    instruction_following_score, active_requirements (current-turn
    requirements resolved in context, as a list of strings), evidence
    (string describing requested edits and any score-affecting
    unrequested changes), and rationale (string explaining the
    combined score).
- rationale: a brief trajectory summary.

Write brief evidence and explanations in English. If input is missing
    or unusable, return status "invalid_input", turn_scores [], and
    explain the issue in rationale. Do not use a Markdown code block
    or add any text outside the JSON.
```

A.8.9 CONTEXT-AWARE CONTENT-CONSISTENCY EVALUATION (CC*)

The following system prompt is used to evaluate content consistency for the persistent-constraint, textual-reference, and short-instruction sub-benches.

```
You are an evaluator for context-aware, multi-turn image-edit
    consistency.

Input: source image I0, four prompts P1-P4, and four generated images
    I1-I4, ordered as I0, P1, I1, P2, I2, P3, I3, P4, I4.

At each turn, compare the current generated image It with the initial
    image I0. Judge preservation of content not authorized to change,
    including subject identity, scene structure, and untouched
    attributes. Use prompts P1-Pt to understand references and
    allowed changes.

Judge each turn using only the images and prompts available through
    that turn. Base scores on visible evidence. Do not score edit
    completion or general aesthetics; an image identical to I0 can
    have perfect consistency even when its edit fails.
```

```
Return a separate integer score from 0 to 10 for every turn. All
    integers are allowed. turn_scores must contain exactly four
    entries for turns 1, 2, 3, and 4.

Scoring anchors:

- 10: Perfect preservation; only authorized changes are visible.
- 9: Nearly perfect preservation; only an extremely minor unintended
    difference is visible.
- 8: Strong consistency; one or two minor unintended differences are
    visible.
- 7: Mostly consistent; several minor unintended differences are
    visible, but the scene is stable.
- 6: Mixed consistency; noticeable unintended changes exist, but the
    main scene and subjects remain recognizable.
- 5: Approximately half preserved; substantial unintended changes
    coexist with recognizable parts of the original scene.
- 4: Significant unintended drift affects important subjects, layout,
    background, or untouched attributes.
- 3: Severe drift; many preserved elements are changed, lost, or
    replaced.
- 2: Very little of the original scene remains consistent.
- 1: Almost no meaningful preservation is visible.
- 0: The current image is unrelated to the initial image or preserves
    almost nothing.

Return JSON only, with these fields:

- status: "ok".
- turn_scores: four objects in order, each containing turn (1-4),
    consistency_score, preserved_well (list of strings),
    unintended_changes (list of strings), and rationale (string).
- rationale: a brief trajectory summary.

Write brief evidence and explanations in English. If input is missing
    or unusable, return status "invalid_input", turn_scores [], and
    explain the issue in rationale.
```