Meta-CoT: Enhancing Granularity and Generalization in Image Editing

Supplementary Material

This is the supplementary material for the paper: "*Meta-CoT: Enhancing Granularity and Generalization in Image Editing.*" We provide the following materials:

1. Theoretical Derivation of Meta-CoT

Here, we provide the proof of the *Theoretical Definition* for Triplet Decomposition presented in Section 3.1 of the main paper. We prove two key results: (1) Triplet Decomposition constrains and simplifies the search space of CoT, thereby reducing the complexity of the editing process; and (2) compared with classical CoT, Meta-CoT achieves a higher level of understanding granularity.

Proof:

Let T denote the original CoT space, and let T_1, T_2, and T_3 represent the editing task, editing object, and required understanding capability, respectively. Then the triplet space is

$$\mathcal{S}_{triplet} = T_1 \times T_2 \times T_3 \tag{1}$$

We further let $\mathcal{S}$ denote the actual single-image editing space (i.e., the valid subset constrained by semantic rules), and let $H = \log|\text{space}|$ represent space complexity (i.e., entropy). According to the subadditivity of entropy and structural constraints, we can prove:

Since semantic rules eliminate numerous invalid combinations (e.g., the object of a "color modification" task cannot be "pose"), the actual valid space satisfies:

$$|\mathcal{S}| \leq |\mathcal{S}_{triplet}| < |\mathcal{T}|, \tag{2}$$

Therefore:

$$\log|\mathcal{S}| \leq \log|\mathcal{S}_{triplet}| = H(T_1, T_2, T_3) < \log|\mathcal{T}| = H(T) \tag{3}$$

This demonstrates that decomposing into task, object, and understanding reduces CoT space complexity, thereby concentrating the probability distribution and lowering editing difficulty.

Next, we use the mutual information per unit entropy $G = \frac{I(T;X_{\text{tgt}})}{H(T)}$ [13] to represent the understanding granularity of CoT. Based on the structural decomposition of mutual information and entropy reduction principle, we can prove:

The information granularity of the triplet is:

$$G(T_1, T_2, T_3) = \frac{I(T_1, T_2, T_3; X_{\text{tgt}})}{H(T_1, T_2, T_3)} \tag{4}$$

Due to the decomposition process, Meta-CoT performs substantially richer and finer-grained reasoning over the three elements of the editing process. For example, in Task Thinking, Meta-CoT performs deeper and more fine-grained reasoning based on the task characteristics (e.g., for style transfer, it explicitly analyzes and articulates the visual attributes of the target style). In Target Traversal, Meta-CoT reasons about the consistency and editing strategy for each target. In contrast, classical CoT (denotes as T) lacks explicit reasoning over these elements and therefore exhibits less information. So:

$$I(T_1, T_2, T_3; X_{\text{tgt}}) = I(T; X_{\text{tgt}}) + \Delta I_{\text{meta}} \tag{5}$$

where $\Delta I_{\text{meta}} > 0$ is the gain from the Meta-CoT decomposition.

So we obtain:

$$\begin{aligned} G(T_1, T_2, T_3) &= \frac{I(T_1, T_2, T_3; X_{\text{tgt}})}{H(T_1, T_2, T_3)} \\ &= \frac{I(T; X_{\text{tgt}}) + \Delta I_{\text{struct}}}{H(T_1, T_2, T_3)} \\ &= \left(1 + \tfrac{\Delta I_{\text{struct}}}{I(T;X_{\text{tgt}})}\right) \tfrac{H(T)}{H(T_1,T_2,T_3)} \tfrac{I(T;X_{\text{tgt}})}{H(T)} \\ &= \underbrace{\left(1 + \tfrac{\Delta I_{\text{struct}}}{I(T;X_{\text{tgt}})}\right)}_{>1} \underbrace{\tfrac{H(T)}{H(T_1,T_2,T_3)}}_{>1} G(T) \\ &> G(T) \end{aligned}$$

That is: $G(T_1, T_2, T_3) > G(T)$, demonstrating that Meta-CoT possesses higher understanding granularity.

2. Human Evaluation

2.1. Human Evaluation Details

We conduct human preference studies on both the 21-task benchmark and ImgEdit, each using 300 samples that cover

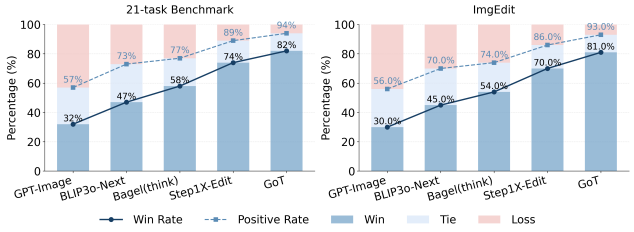

Figure 1. **Human Study.** Pairwise human comparison between other methods and ours on the 21-task benchmark and ImgEdit.

various editing tasks. The evaluation involves 30 participants (ages 20–55, 15 males and 15 females). They come from diverse disciplinary backgrounds, including computer science, art, design, graphics, engineering, and social sciences. They possess basic literacy in visual quality assessment and have not previously been exposed to our model.

To construct the evaluation set, we randomly sample images and instructions from the two benchmarks while ensuring balanced task coverage across all task types included in both benchmarks. For each sample, we generate pairwise comparisons between our method and baseline outputs. All image pairs are double-blind randomized: the left–right ordering of the two edited results is shuffled independently for every trial, and model identities are hidden from participants.

During each trial, participants are shown the input image, the text instruction, and two edited results. They are asked to choose the version that better matches (1) instruction faithfulness and (2) visual/aesthetic quality. When both results are judged comparable, participants may select "tie." Each judgment is recorded as a win, loss, or tie for our method, from which we compute the win rate (wins / total) and positive rate (wins + ties). We also include basic quality-control procedures: 10% of the trials are randomly repeated to measure intra-annotator consistency, and inconsistent annotators ($> 20\%$ disagreement) are excluded.

2.2. Human Evaluation Results

As shown in Figure 1, on the 21-task benchmark, Meta-CoT achieves a win rate of 58%, outperforming Bagel (think) (23%) by 35 points, demonstrating the effectiveness of our approach. Our method also achieves a win rate of 47%, outperforming the current best open-source unified model, BLIP3o-Next (27%), by 20 points.

3. Implementation Details

3.1. Training Details

During the SFT stage, we train the model for 10k steps on 48×H20 GPUs. All training is performed in bfloat16 mixed precision for memory efficiency and stable optimization. Optimization uses AdamW with a learning rate of 1e-4, cosine decay, and a 100-step warmup. During the RL stage, we further fine-tune the model for 500 steps on an additional 20k high-quality editing dataset using 32×H20 GPUs. The learning rate is set to 1e-6, and the group size is 16 for each prompt.

3.2. VLM-based Evaluation Details

We strictly follow the evaluation protocols in [2, 7, 9, 14], using GPT-4.1 to compute all metrics, ensuring fairness and accuracy. On the 21-task benchmark, we follow GEdit-Bench [9] and adopt VIEScore [5] as our evaluation metric. VIEScore is a widely used, human-aligned metric in image editing that evaluates performance from four aspects: instruction following, subject consistency, naturalness, and artifacts, and computes the geometric mean of these four dimensions as the overall score.

To further improve the stability and reliability of the metrics, each output is scored by GPT-4.1 5 times, and the mean of these repeated evaluations is used as the final score. This procedure ensures the robustness of our automated evaluation.

4. Data / Reward Construction Details

4.1. Editing Image Instruction Pairs Construction

We source high-quality source images from publicly available datasets [4, 11]. After obtaining the source images, we first define a diverse edit taxonomy and employ Gemini-2.5-Flash to uniformly and randomly generate a broad range of candidate instructions for each image based on its content and editing task type. Subsequently, we implement a filtering mechanism that combines automated scoring with human review to select the most appropriate and high-quality instructions, ensuring the final instruction set is both diverse and aligned with the intended editing tasks. Next, we leverage [6, 10, 14] to generate edited images from the source images and instructions. We then compute VIEScore for each generated image-instruction pair using VLMs, retaining only samples with perfect scores ($sc\_score = 10$ and $pq\_score = 10$). To mitigate model-specific biases, we perform independent screening rounds using both Gemini-2.5-Flash and GPT-4.1. Finally, we conduct manual quality control through sampling inspection of each generation batch to ensure the quality of the generated data.

4.2. Meta-CoT Construction

In Section 3.6 of the main paper, we introduce the construction pipeline of Meta-CoT. Leveraging Qwen2.5 [15] and Qwen2.5-VL [1], we sequentially generate the task type and the Meta-CoT, where the three components of Meta-CoT (Meta-Task Summary, Task Thinking, and Target Editing Mode Traversal) are produced step-by-step based on the input image-editing pair and instruction. Both the predicted task type and the generated Meta-CoT are further validated

by Gemini-2.5-Flash. We intentionally use a VLM different from the one used for generation to avoid preference bias.

For future research, we provide all prompts used in constructing Meta-CoT. Tables 4 and 5 present the prompts guiding the VLM to infer the editing task type from the input instruction (the prompt is long and thus split into two tables). These prompts also include our definitions and examples of each task type. Table 6 shows the prompts used to generate the Task Thinking and Target Editing Mode Traversal given an input image pair and editing instruction. The prompt guides the VLM to generate traversal over all editing targets and their editing modes. It also includes several example of prompts that guides the VLM to generate task-specific reasoning processes. Table 7 provides the prompt used to decompose an input editing instruction into one or more meta-tasks. This prompt contains our definitions of meta-tasks as well as illustrative examples of meta-task decomposition.

4.3. CEC Reward Construction

In Table 8, we present the prompt used to generate the CoT-Editing Consistency (CEC) Reward. In this prompt, the VLM is requested to assess whether the produced edit is consistent with both the task-level thinking and the target editing mode traversal. The model ultimately outputs a score along with an explanation. Consistency checking with task-level reasoning enables the VLM to evaluate the correctness of the edit from a global perspective, while checking consistency across the editing modes of all target subjects allows the VLM to assess accuracy at a finer granularity. Together, these two dimensions allow the CEC Reward to provide a precise evaluation of the semantic correctness of image editing.

5. Additional Methodological Details

5.1. Group Relative Policy Optimization

In GRPO [12], given a task question, the model generates a set of N potential responses $\{O_1, O_2, \ldots, O_N\}$. Each response is evaluated by taking the corresponding actions and computing its reward $\{R_1, R_2, \ldots, R_N\}$. Unlike PPO, which relies on a single reward signal and a critic to estimate the value function, GRPO normalizes these rewards to calculate the relative advantage of each response. The relative quality A_i of the i-th response is computed as

$$A_i = \frac{r_i - \text{Mean}(\{r_1, r_2, \ldots, r_N\})}{\text{Std}(\{r_1, r_2, \ldots, r_N\})},$$

where Mean and Std represent the mean and standard deviation of the rewards, respectively. This normalization step ensures that responses are compared within the context of the group, allowing GRPO to better capture nuanced differences between candidates. Policy updates are further constrained by minimizing the KL divergence between the updated and reference models, ensuring stable RL learning. Refer to [3, 12] for more details.

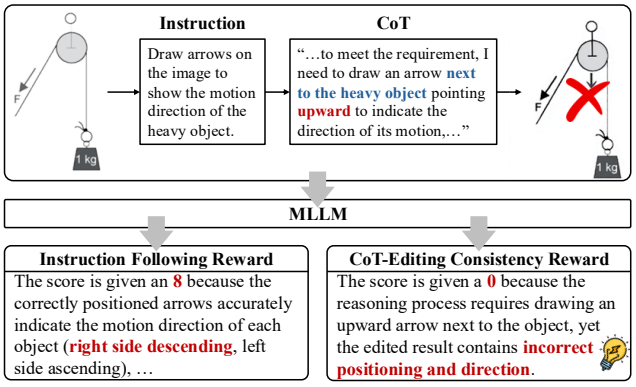

Figure 2. **Comparison between Instruction Following and CEC Reward based on Gemini-2.5-Flash.** CEC Reward better reflects editing semantic accuracy than Instruction Following.

Table 1. **Comparison of CEC and IF Rewards on the 21-Task Benchmark After GRPO Training with different rewards.**

Method	IF Reward	CEC Reward
SFT(Meta-CoT)	7.23	6.27
+ RL(IF Reward)	7.35	6.94
+ RL(CEC Reward)	**7.44**	**8.03**

6. Additional Experimental Results

6.1. CEC Reward vs Instruction Following

In this section, we compare the reward effects of the CoT-Editing Consistency Reward (CEC Reward) and the Instruction Following Reward (hereafter referred to as IF Reward) to demonstrate that the CEC Reward better reflects semantic accuracy. The Instruction Following Reward we compare against is derived from VIEScore [5], which similarly leverages MLLMs to evaluate instruction-following ability.

As shown in Figure 2, the CEC Reward reflects the semantic accuracy of edits more effectively than the IF Reward. This is because instructions are often highly condensed representations of the editing intent, containing implicit and rich editing operations. Through CoT, these operations can be decomposed in detail. Therefore, evaluating the consistency between the edit and the CoT provides a more faithful measure of semantic accuracy than directly comparing the edit with the instruction.

To further validate the advantage of the CEC Reward over the IF Reward, we performed GRPO [8, 12] training on the model fine-tuned with Meta-CoT, using either CEC Reward or IF Reward as the optimization reward. We then compared the changes in model performance under both

Table 2. **Ablation study on the number of meta-tasks defined and tasks trained (Full Task Metrics).** All metrics are evaluated using GPT-4.1. Train Editing Only denotes the setting trained with the same parameters and editing data as our method, but without Meta-CoT. *(n meta)* denotes defining n meta-tasks and training only on them. *(5 meta, full-task)* indicates defining 5 meta-tasks, training on full tasks, and decomposing each task's data into meta-tasks. We mark the five meta-tasks selected for the final setting with *.

Method / Task	Background	Color	Material	Action	Human Attribute	Style	**Add***	**Remove***	**Replace***	Text	Tone
Train Editing Only	6.743	6.537	6.052	4.155	4.257	6.571	7.363	6.296	6.687	2.651	6.547
SFT(3 meta)	7.024	6.611	6.214	4.153	4.355	6.705	7.606	7.795	6.913	2.886	6.663
SFT(4 meta)	7.041	6.966	6.238	4.218	4.434	6.769	7.613	7.802	6.918	2.905	6.857
SFT(5 meta)	7.128	7.136	6.354	4.365	4.580	6.837	7.612	7.813	6.924	3.143	7.064
SFT(6 meta)	7.132	7.140	6.393	4.408	4.632	6.892	7.616	7.810	6.917	3.162	7.056
SFT(5meta, full-task)	7.167	7.197	6.469	4.452	4.730	6.932	7.627	7.821	6.934	3.235	7.192

Method / Task	Causal	Logical	Spatial Reasoning	Temporal	**Camera***	Structure	**Position***	Quantity	Specified Quantity	Multi-Instruction	**Average**
Train Editing Only	5.710	3.217	4.683	5.663	5.964	5.629	5.177	4.957	5.093	6.349	5.538
SFT(3 meta)	5.805	3.212	4.694	5.733	6.034	5.899	5.247	5.627	5.243	6.219	5.745
SFT(4 meta)	6.174	3.219	4.712	5.983	6.790	6.386	5.173	6.025	5.560	6.517	5.919
SFT(5 meta)	6.450	3.371	4.757	6.075	6.827	6.993	5.843	6.262	5.750	6.695	6.094
SFT(6 meta)	6.485	3.395	4.788	6.102	6.861	7.060	5.850	6.315	5.812	6.745	6.122
SFT(5meta, full-task)	6.617	3.476	4.820	6.186	6.883	7.264	5.866	6.432	5.980	6.895	6.199

rewards before and after training. All experiments used the same training data, random seeds, hyperparameters, and number of training steps. As shown in Table 1, after 500 steps of GRPO training, the setting optimized with CEC Reward achieved greater improvements on both CEC and IF metrics. In contrast, the setting optimized with IF Reward also improved on both metrics, but the gains were smaller, particularly on the CEC Reward.

6.2. Generalization Capability of Meta-CoT

Due to space limitations, the main paper reports only the averaged results for the ablation study on the number of meta-tasks defined and tasks trained. Therefore, in Table 2, we provide the complete results across all 21 tasks. From these results, we observe the following:

First, training solely on the five selected meta-tasks (highlighted in bold with * annotations in the table) already enables strong generalization to the remaining 16 unseen tasks. On these unseen tasks, the 5-meta setting consistently outperforms the Train Editing Only baseline and achieves performance close to the 5-meta, full-task setting, where all tasks are included during training. This demonstrates that Meta-task Decomposition endows the model with strong generalization capabilities across diverse editing scenarios.

Additionally, the results further justify our choice of five meta-tasks: When reducing the number of meta-tasks from five to three or four, we observe a substantial drop in overall editing performance. Conversely, increasing the number from five to six yields little improvement.

6.3. Performance on the RiseBench

Our 21-task benchmark includes RiseBench [16], and we evaluate the entire benchmark using VIEScore. Since RiseBench provides its own evaluation metrics specifically designed for reasoning-centric editing tasks, we additionally report the results on RiseBench separately for rigor and fairness. Following the official evaluation protocol of RiseBench, we use GPT-4.1 as the evaluation model. As shown in Table 3, our method achieves a 23.5% improvement over Bagel (w/CoT) and reaches the performance level of commercial models such as Gemini-2.0-Flash-exp.

Table 3. Comparison of models on four reasoning dimensions (Temporal, Causal, Spatial, Logical) and Overall performance.

Model	**T. (%)**	**C. (%)**	**S. (%)**	**L. (%)**	**O. (%)**
Closed-source Models					
FLUX.1-Kontext-Dev	2.3	5.5	13.0	1.2	5.8
Gemini-2.0-Flash-pre	10.6	13.3	11.0	2.3	9.4
Seedream-4.0	12.9	12.2	11.0	7.1	10.8
Gemini-2.0-Flash-exp	8.2	15.5	23.0	4.7	13.3
GPT-Image-1-mini	24.7	28.9	33.0	9.4	24.4
GPT-Image-1	34.1	32.2	37.0	10.6	28.9
Gemini-2.5-Flash-Image	25.9	47.8	37.0	18.8	32.8
Open-source Models					
FLUX.1-Canny	0.0	0.0	0.0	0.0	0.0
HiDream-Edit	0.0	0.0	0.0	0.0	0.0
EMU2	1.2	1.1	0.0	0.0	0.5
OmniGen	1.2	1.0	0.0	1.2	0.8
Step1X-Edit	0.0	2.2	2.0	3.5	1.9
Ovis-U1	1.2	3.3	4.0	2.4	2.8
BAGEL	2.4	5.6	14.0	1.2	6.1
Qwen-Image-Edit	4.7	10.0	17.0	2.4	8.9
BAGEL (w/ CoT)	5.9	17.8	21.0	1.2	11.9
Meta-CoT + RL (Ours)	11.8	18.9	22.6	2.5	14.7

6.4. Additional Qualitative Results

In Figures 3 and 4, we provide additional qualitative results covering a broader range of task types. As shown, our method consistently demonstrates substantial improvements over the baseline approaches across diverse categories of editing tasks.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 2

[2] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025. 2

[3] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 3

[4] Ivan Krasin, Tom Duerig, Neil Alldrin, Andreas Veit, Sami Abu-El-Haija, Serge Belongie, David Cai, Zheyun Feng, Vittorio Ferrari, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, Kevin Murphy, Dhyanesh Narayanan, Saurabh Shetty, Yang Song, Joseph Tighe, Andrea Vedaldi, Sudheendra Vijayanarasimhan, and Oriol Vinyals. Open-images: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available fromn-https://storage.googleapis.com/openimages/web/index.html*, 2017. https://storage.googleapis.com/openimages/web/factsfigures.html. 2

[5] Max Ku, Dongfu Jiang, Cong Wei, Xiang Yue, and Wenhu Chen. Viescore: Towards explainable metrics for conditional image synthesis evaluation. *arXiv preprint arXiv:2312.14867*, 2023. 2, 3

[6] 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*, 2025. 2

[7] Bin Lin, Zongjian Li, Xinhua Cheng, Yuwei Niu, Yang Ye, Xianyi He, Shenghai Yuan, Wangbo Yu, Shaodong Wang, Yunyang Ge, et al. Uniworld-v1: High-resolution semantic encoders for unified visual understanding and generation. *arXiv preprint arXiv:2506.03147*, 2025. 2

[8] Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025. 3

[9] 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. 2

[10] OpenAI. Gpt-image-1, 2025. 2

[11] 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. *NeurIPS*, 35:25278–25294, 2022. 2

[12] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 3

[13] Henri Theil. On the estimation of relationships involving qualitative variables. *American Journal of Sociology*, 76(1): 103–154, 1970. 1

[14] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025. 2

[15] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. 2

[16] Xiangyu Zhao, Peiyuan Zhang, Kexian Tang, Xiaorong Zhu, Hao Li, Wenhao Chai, Zicheng Zhang, Renqiu Xia, Guangtao Zhai, Junchi Yan, et al. Envisioning beyond the pixels: Benchmarking reasoning-informed visual editing. *arXiv preprint arXiv:2504.02826*, 2025. 4

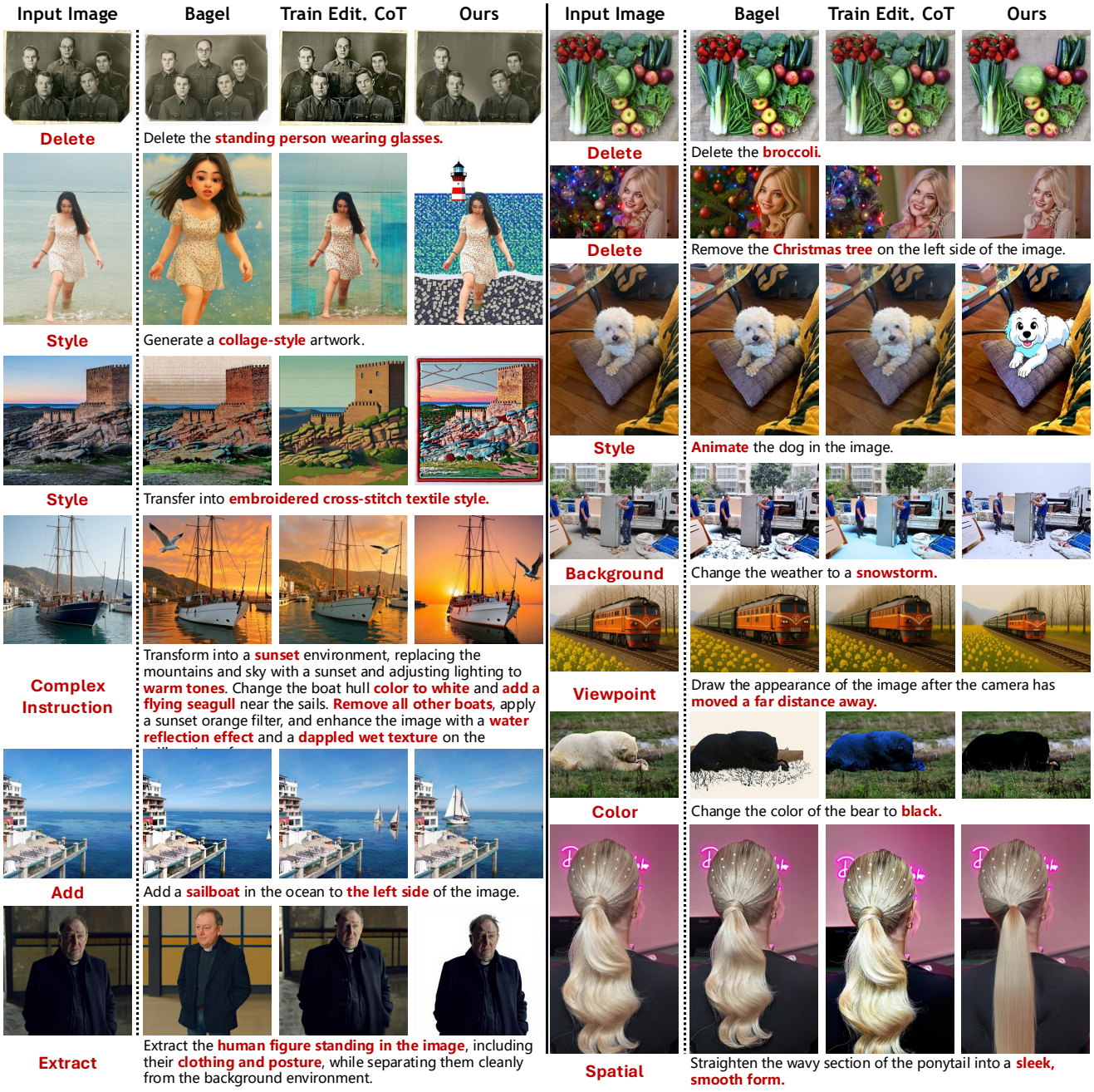

Figure 3. **Qualitative results across diverse editing tasks**, including conventional editing, reasoning-based editing, and multi-instruction editing (Zoom in to view).

Figure 4. **Qualitative results across diverse editing tasks**, including conventional editing, reasoning-based editing, and multi-instruction editing (Zoom in to view).

###[System Role Instruction]
You are an expert visual-editing classifier. Read the input English instruction about an image, then output exactly one label (no punctuation or extra words) that best matches the requested edit type.

###[Rules]
1. Read the entire instruction carefully; ignore secondary, decorative words.
2. Identify the one operation that is explicitly commanded.
3. Return only the label—no extra text, punctuation, or reasoning.
4. If more than one editing operation is mentioned, then it is a complex instruction. In this case, please output the label in the format: Complex Instruction (Type1, Type2, . . .) listing every task type detected, in the order they appear.

###[Label Definitions]
- Object Addition: Command adds a new, previously non-existent object, animal, person, or text element into the scene.
  - Keywords/Examples: “add”, “insert”, “put in”, “place”.
- Object Removal: Command removes an existing object, creature, or text element.
  - Keywords/Examples: “remove”, “delete”, “erase”, “take out”, “eliminate”.
- Object Replacement: Command swaps one existing object for another different object while keeping its approximate location and scale.
  - Keywords/Examples: “replace”, “swap”, “change . . . into . . . ”, “turn . . . into . . . ”.
- Style Transfer: Command changes the overall artistic style—e.g., to oil-painting, sketch, cyber-punk—without altering content geometry.
  - Keywords/Examples: “in the style of”, “as a watercolor”, “render as Disney style”, “make it look like the style of ...”.
- Camera Motion: Command shifts the virtual camera viewpoint: zoom, pan, dolly, orbit, rotate, tilt.
  - Keywords/Examples: “zoom in/out”, “pan left/right”, “rotate view”, “move camera”.
- Text Editing: Command specifically modifies textual content already present in the image or adds new text.
  - Keywords/Examples: “change the text to”, “write . . . on the sign”, “correct the spelling”.
- Shape Modification: Command alters the geometry or silhouette of a single existing object (stretch, shrink, twist, round).
  - Keywords/Examples: “make it taller”, “elongate”, “bend”, “curve”, “squash”, “inflate”.
- Structural Change: Command rearranges high-level scene layout or architecture—walls, doors, buildings, etc.
  - Keywords/Examples: “rebuild”, “rearrange room layout”, “open the wall”, “add a second floor”.
- Temporal Change: Command shifts the depicted moment to date, season, historical era or a different time of day (dawn, noon, dusk, night) without altering the scene’s content or layout.
  - Keywords/Examples: “set it at sunset”, “winter version”, “make it 1920s”, “what will it look like in 2025”, “what will it look like at noon”, “what will it look like at night”, etc.
- Causal Change: Changes resulting from an interaction between the subject and external environment, or the natural change of the subject itself, e.g., “what the zipper looks like when zipped up,” “the car after it has been hit,” “Show the lotus flower in full bloom”, etc.
  - Keywords/Examples: “what will it look like when zipped up”, “what will it look like after it has been hit”, “what will it look like in full bloom”, etc.
- Logical Reasoning: Instruction requires logical deduction from visual information to determine the specific edit operation (e.g., complete a visual pattern, solve a graph-based puzzle, draw the next move in chess, derive the next step in a physics/chemistry diagram). The model must infer how to edit the image based on rules or patterns evident in the image.
  - Keywords/Examples: “complete the next shape in the sequence”, “solve the puzzle”, “draw the next move in chess”, “add the missing resistor to minimize current in the circuit”, etc.
- Color Change: Command alters the color of an existing object or region.
  - Keywords/Examples: “turn it red”, “make the sky purple”, “change shirt color to blue”.
- Quantity Change: Command increases or decreases the count of a class of objects.
  - Keywords/Examples: “more”, “fewer”, “add some”, “reduce the number of”, “set the count to”, “add two more”, etc.

Table 4. Prompt for Meta-CoT generation (Task Type inference) [part 1].

- Specified Quantity Change: Instruction singles out exactly N instances of a subject that already appears in multiples and performs any edit (add, delete, move, recolor, etc.) on precisely those N instances. The key is the explicit mention of the exact count to be acted upon.
  - Keywords/Examples: “move three birds to the lower branch”, “recolor exactly two of the five apples to green”, etc.
- Positional Relationship Change: Command moves an existing object to a new absolute or relative location or swaps the positions of multiple existing objects.
  - Keywords/Examples: “move the apple to the left”, “place behind the car”, “shift upward”, “swap positions of A and B”, etc.
- Complex Instruction: Instruction contains multiple editing operations.
  - Keywords/Examples: “add a red umbrella next to the bench, and remove the green apple”, etc.
- Tone Adjustment: Apply global, non-object-specific adjustments to contrast, saturation, or overall color tone of the entire image without altering individual object colors or outlines.
  - Keywords/Examples: “convert to high-contrast”, “desaturate to grayscale”, “shift to warm/cool tone”, etc.
- Human Attribute Modification: Modify any human attribute other than clothing, age, or motion, such as facial expression, hairstyle, hair color, skin tone, body weight, or height, without altering the person’s garments or apparent age.
  - Keywords/Examples: “change the hairstyle to a bob”, “change the man’s expression to a big smile”, “change the body weight to 100kg”, “change the man to a woman”, etc.
- Material Change: Alter the material or texture of an object without changing its shape, geometry, or outline.
  - Keywords/Examples: “change the material of the apple to plastic”, “change the table from wood to polished marble”, “turn the metal kettle into frosted glass,” etc.
- Motion Change: Alter the pose or motion of a person or animal depicted in the image without changing identity, appearance, or surrounding elements.
  - Keywords/Examples: “change the runner from standing still to mid-stride”, “make the cat leap instead of sitting”, etc.
- Background Change: Replace or alter the entire background while keeping the foreground subjects unchanged.
  - Keywords/Examples: “change the background to a forest”, “change the background to a beach”, “change the background to a city”, “replace the indoor studio with a beach sunset”, etc.

###[Output Example]

Example 1:
Input: “Place a red umbrella next to the bench.”
Output: Object Addition

Example 2:
Input: “Change the man to a woman and add a red umbrella next to the bench.”
Output: Complex Instruction (Human Attribute Modification, Object Addition)

###[Input Editing Instruction]

[Place the editing instruction here]

Table 5. Prompt for Meta-CoT generation (Task Type inference) [part 2].

###[System Role Instruction]

You are a professional and creative image editor. Your task is to generate a fine-grained, detailed editing instruction based on:
1) the source image (first image),
2) the edited image (second image),
3) the simple editing instruction,
4) the editing task type.

###[Workflow]

- First, repeat the editing instruction, like "The editing instruction is: [instruction]".
- Next, perform task-specific reasoning based on the type of task:
• If the task type includes "Object Addition", provide a detailed description of the object's appearance and the exact placement for insertion, and reason about whether, and how it will interact with existing objects in the image.
• If the task type includes "Object Removal", reason about what the region previously occupied by the removed object should look like after its removal. • If the task type includes "Style Transfer", describe the target style's visual characteristics in objective, observable terms (e.g., composition, color, line, texture, form).
• If the task type includes "Camera Motion", describe the global viewpoint transformation and any subjects that newly appear, disappear, or change in visibility after the shift.
• If the task type includes "Material Change", describe the target material's visual characteristics in objective, observable terms (e.g., gloss level, roughness, color variation, texture scale).
• If the task type includes "Temporal Change", "Causal Change", or "Logical Reasoning", provide the detailed reasoning required to produce the edited image, transforming the editing instruction into concrete and executable editing operations.
...
- Finally, identify all elements and subjects in the image, and describe whether each element or subject needs to be edited and how to edit it as detailed as possible.

###[Notes]

- Output the fine-grained editing instruction directly without any other text.
- There is no need to describe the original content of each region; only the required edits need to be described.
- Describe how you will edit the image, but do not mention any editing tools.
- Do not mention any content that does not exist in the source image or the edited image.
- For the elements and subjects that need to be edited, please carefully observe the differences between them in the source image and the edited image, and describe how to edit them in detail.
- For the editing description, please be as specific and accurate as possible, ensuring that there is no ambiguity.
- If the edited image has elements that do not exist in the original image, please describe the appearance of the new elements in detail.
- Do not use "or" or other ambiguous expressions.
- Do not mention the edited image in the output.
- Do not use "delicate", "natural", "realistic", or other abstract words to describe the editing process.

###[Input Simple Editing Instruction]

[Place the editing instruction here]

Table 6. Prompt for Meta-CoT generation (Task Thinking (several examples) and Target Editing Mode Traversal).

###[System Role Instruction]

You are a professional and logical image editor. Your task is to reformat an existing fine-grained editing instruction into a structured reasoning process with explicit steps based on the source image, the edited image, and the editing instruction.

###[Workflow]

Infer the editing process and summarize it as a sequence of the allowed primitive editing operations only:
a. Addition: add a new subject or a specific attribute of a subject into the scene.
b. Deletion: remove an existing subject or a specific attribute of a subject from the scene.
c. Replacement: replace an existing subject or a specific attribute of a subject (such as style, tone, shape, color, human attribute, material, action, etc.).
d. Camera movement: perspective changes such as zoom, pan, dolly, orbit, rotate, tilt.
e. Position transformation: move an existing subject to a new absolute or relative location or swap the positions of multiple existing subjects.

###[Output Example]

Example 1:
Input source image: a blackboard with "1+1=?" written on it.
Input editing instruction: "Fill in the answer on the blackboard."
Reasoning: The blackboard shows "1+1=?". Since 1+1=2, to complete this task the edit can be summarized as: *Replace the question mark with the number 2.*

Example 2:
Input editing instruction: "Remove the dog in the image and change the color of the sky to blue."
Reasoning: To complete this task, the edit can be summarized as: *Delete the dog in the image. Replace the sky's color with blue.*

Example 3:
Input editing instruction: "Make the cat's fur white."
Reasoning: To complete this task, the edit can be summarized as: *Replace the color of the cat's fur with white.*

Example 4:
Input source image: a small sapling
Input editing instruction: "What will it look like in ten years?" Reasoning: In ten years, the sapling will have grown into a large tree. Therefore, the edit required to fulfill this instruction can be summarized as: *Replace the small sapling with a large tree.*

###[Notes]

- If the instruction is vague or requires logical deduction, expand the reasoning to produce precise, unambiguous editing operations.
- Only reformat the input instruction into the structure above; do not invent new edits.
- Keep the wording precise and avoid ambiguous expressions.
- Do not describe the original content of regions; only describe the required edits. - Do not mention editing tools or the edited image.
- Do not use vague terms like "delicate," "natural," or "realistic."
- The summary of the editing process must consist of the primitive editing operations listed above. For example: 'Replace the style of the image with Art Nouveau. Add a stream of yolk. Swap the position of the cat and the dog.' etc.
- The summary of the editing process must be concise and clear.

###[Input Fine-Grained Editing Instruction]

[Place the editing instruction here]

Table 7. Prompt for Meta-CoT generation (Meta-task Summary).

###[System Role Instruction]

You are a rigorous editing quality inspector.
Task: Rate the consistency between the "editing thinking process" and the final edited image on a 0-10 scale, and provide a concise reason.

###[You will be given the following inputs]

1. Original image
2. Edit instruction (user's short request)
3. Editing thinking process (two parts):
- Task thinking: A detailed and task-specific analysis of the editing intent.
- Element-wise plan: an itemized list of every visible subject/element, each with:
  - Edit decision (whether to edit the subject or not)
  - Edit details (how to edit the subject)
4. Final edited image

###[Scoring rules]

Check every point:
1. Task thinking: the edited image must fully realize the task thinking described in 3-a.
2. Element-wise plan:
- For every listed subject/element, verify that both the edit decision and the edit details are executed exactly as stated.
- Deduct points for any of the following:
  - A subject that should remain unchanged has been altered;
  - A subject that should be edited has not been changed;
  - A subject that should be edited has been changed, but the direction or extent of the change does not match the description.

###[Notes]

Only judge whether the edited image is consistent with the editing thinking process.

###[Output Format]

Output the score and reason in JSON format, no extra text.
{
"score": [int 0-10],
"reason": "[$\leq$ 50-char English explanation]"
}

###[Input Editing Instruction]

[Place the editing instruction here]

###[Editing Thinking Process]

[Place the thinking process here]

Table 8. Prompt for CoT-Editing Consistency Reward generation.