MIRA: Multimodal Iterative Reasoning Agent for Image Editing Supplementary Material

Ziyun Zeng[1] , Hang Hua[2] , Jiebo Luo[1]
[1]University of Rochester, [2]MIT-IBM Watson AI Lab
ziyun.zeng@rochester.edu, hang.hua1@ibm.com, jluo@cs.rochester.edu

A. Dataset Details

A.1. Data Construction Pipeline

To enable MIRA 's iterative perception–reasoning–action framework, we construct MIRA-EDITING, a large-scale dataset comprising 150K high-quality instruction-image editing trajectories. Figure A illustrates the overall data construction pipeline, which consists of three major stages: data preprocessing and rewriting, candidate generation and ranking, and final sample construction.

We begin with the multi-turn editing sequences provided in Seed-Data-Edit [5] and ImgEdit [10], where each sample contains several atomic edits expressed across multiple conversational turns. To transform these into training samples suitable for iterative reasoning, we first perform instruction aggregation, merging all atomic edit descriptions into a single complex instruction. To increase compositional diversity, we generate both in-order and permuted variants of the aggregated instruction, encouraging the model to learn order-invariant reasoning over multi-step edit descriptions. To further enhance linguistic richness, we apply a two-level rewriting strategy. First, each atomic instruction is paraphrased independently to introduce linguistic variation; then the full complex instruction is holistically rewritten, altering sentence structure, connectives, and phrasing while preserving semantic meaning. This process yields multiple rewritten instruction variants per editing sequence, enabling MIRA to generalize across the diversity in real-world user instructions.

For each rewritten instruction, we execute the instruction using Flux.1-Kontext, which a strong open-source instruction-guided image editing model. This produces multiple candidate edited trajectories, each representing an image-editing outcome conditioned on a different paraphrased instruction. We use Gemini-2.5-Flash and Qwen2.5-VL-72B-Instruct as evaluators, to compute semantic consistency between the edited image and the original complex instruction. We then rank all candidates and retain the top-1 highest-scoring edited image as a sample of the dataset, ensuring high fidelity and alignment.

Each selected editing sample is converted into step-wise supervision tuples suitable for training MIRA. Specifically, we decompose the final editing trajectory into 3 types of samples, as shown in Figure 3 in the main paper. Each dataset entry contains the original image I_0, last round edited result I_{t-1}, the complex instruction C, and the corresponding atomic instruction u_t, providing expressive multimodal supervision for iterative reasoning and editing.

A.2. Dataset Statistics

Figure B summarizes key statistics of the MIRA-EDITING dataset.

B. Additional Training Details

B.1. Hyperparameters

MIRA is implemented by PyTorch, and we adopt Qwen2.5-VL-7B-Instruct and Qwen2.5-VL-3B-Instruct as the backbone for MIRA 7B and MIRA 3B respectively.

Stage 1: We follow the standard self-supervised tuning (SFT) settings provided by Qwen official codebase [1]. We train the model for 1300 steps using the AdamW optimizer with a learning rate of 1.0×10^{-5} and a cosine decay scheduler with a 0.1 warmup ratio with 150k training data. The input image resolution is fixed to 512.

Stage 2: In the Reinforcement Learning stage, we apply Group Relative Policy Optimization (GRPO) to further align MIRA with editing objectives. To ensure training stability and computational efficiency, we adopt Low-Rank Adaptation (LoRA) for the policy model update. We train for 200 steps. For each input instruction, we sample a group of $G = 5$ candidate outputs to estimate the baseline. The learning rate is set to 5×10^{-6}, and the KL divergence coefficient β is set to 0.04. Our customized reward function weights in Equation (5) in the main paper are empirically set to $\lambda_{sc} = 2.0$ and $\lambda_{pq} = 0.5$, prioritizing semantic alignment while maintaining perceptual quality.

[1]https://github.com/QwenLM/Qwen3-VL

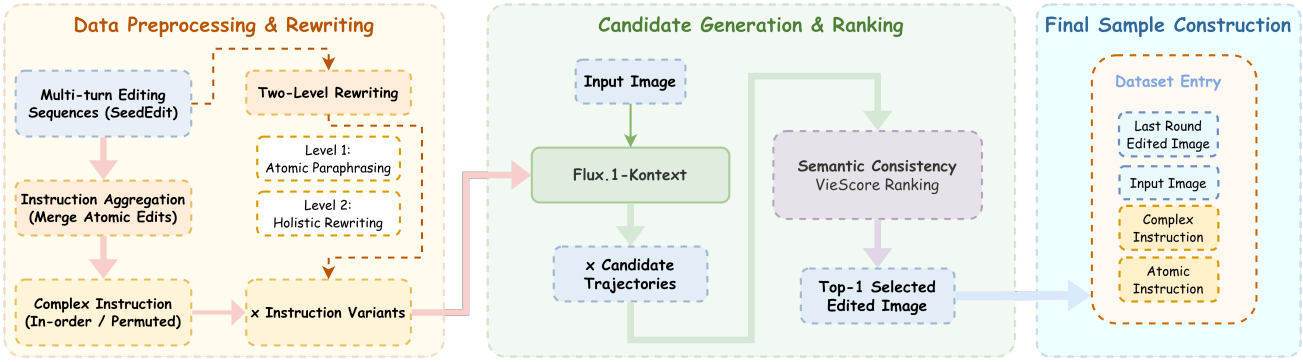

Figure A. **Overview of the MIRA-EDITING Data Construction Pipeline.** The dataset is built through three stages: (1) Data Preprocessing & Rewriting, where multi-turn atomic edits from org multi-turn data samples are merged into complex instructions and expanded via two-level rewriting; (2) Candidate Generation & Ranking, where an open-source image editing model executes each rewritten instruction and use viescore to select the most semantically aligned edited result; and (3) Final Sample Construction, which converts the selected trajectory into step-wise supervision including the input image, last round edited image, complex instruction, and corresponding atomic instruction.

.

B.2. System Prompts

Instruction Template

```
You are an image editing instruction
planner. Your task: Based on the
original image, the current edited
image, and the final instruction
set, determine the next image editing
action that should be applied to move
the image closer to the desired final
result.

Inputs:
Original Image: <image>
Latest Edited Image: <image>
Instruction Set: <instruction>

Instructions:
1. Compare the original and latest
edited image.
2. Determine which part(s) of the
instruction set have already been
applied.
3. Identify the next most important
instruction that has not been
completed.
4. Output the next editing
instruction clearly and specifically.

Output format:
NEXT_EDITING_INSTRUCTION: <short,
clear command to be applied next>
```

C. Additional Experiment Results

C.1. Effectiveness on Simple Instructions

To assess the effectiveness of MIRA on simple instructions that does not require multi-step reasoning, we conduct experiments on single-step subset of Imgedit-Bench [10]. Suprisingly, Table A reflects that integrating MIRA with InstructPix2Pix yields consistent improvements across almost every editing types, indicating that MIRA can also operate as an instruction refiner, providing clearer, more structured instructions even if the task itself requires only a single transformation. This demonstrates that MIRA not only experts in decomposing complex instructions into atomic operations by reasoning but can also serve as an instruction refiner, generating clearer and more model-friendly editing instructions.

C.2. Ablations on Reasoning Steps and Excution Quality

To evaluate whether a larger reasoning budget improves execution quality, we vary the maximum number of inference steps from 3 to 7 and report the corresponding performance in Table B. Across both semantic consistency and perceptual quality metrics, we observe only minor variations as the maximum step budget increases. Allowing additional steps yields modest improvements, most notably in semantic consistency, suggesting that MIRA occasionally benefits from the extra reasoning capacity when handling more challenging edits. However, the overall changes remain small, indicating that iterative reasoning depth is not the primary factor governing final output quality.

Table A. **Quantitative Comparison on Imgedit-Bench single-step editing tasks.** We evaluate whether MIRA can act as an effective instruction refiner when plugged-and-played with an open-source image editing model.

Method	Add	Adjust	Extract	Replace	Remove	Background	Style	Action	Overall
InstructPix2Pix [3]	2.19	2.87	3.01	2.10	1.66	2.00	2.19	1.81	2.29
InstructPix2Pix + MIRA	2.73$_{+24.66\%}$	3.17$_{+10.45\%}$	3.79$_{+25.91\%}$	1.75$_{-16.67\%}$	2.15$_{+29.52\%}$	2.14$_{+7.00\%}$	2.49$_{+13.70\%}$	2.56$_{+41.44\%}$	2.63$_{+14.85\%}$

Table B. **Ablation Study on the Impact of Maximum Inference Steps on MIRA 's Editing Performance.** We evaluate Flux.1-Kontext + MIRA under different maximum number of inference step (from 3 to 7) and report semantic consistency and perceptual quality across multiple metrics. The **best** and second best results are in bold and underlined, respectively.

Method	GPT-SC ↑	Gemini-SC ↑	Qwen3VL-SC ↑	EditScore-SC ↑	ARNIQA ↑	TOPIQ ↑	EditScore-PQ ↑	EditScore-OA ↑
Flux.1-Kontext + MIRA + Max=3	5.912	6.626	5.788	**8.659**	0.611	0.346	**5.550**	**6.785**
Flux.1-Kontext + MIRA + Max=4	6.204	6.284	5.800	8.553	0.614	0.350	5.410	6.654
Flux.1-Kontext + MIRA + Max=5	6.202	6.670	5.802	8.532	**0.619**	**0.353**	5.473	6.610
Flux.1-Kontext + MIRA + Max=6	6.304	**6.760**	5.840	8.487	0.617	0.352	5.406	6.628
Flux.1-Kontext + MIRA + Max=7	**6.320**	6.730	**5.892**	8.480	0.618	**0.353**	5.337	6.556

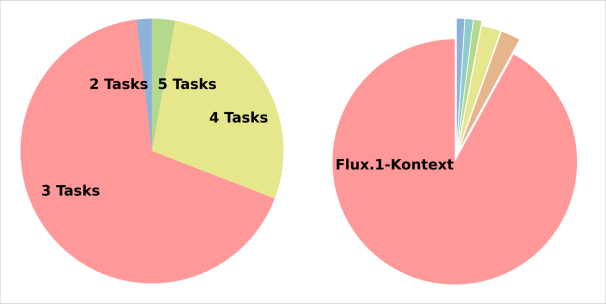

Figure B. **Dataset statistics for MIRA-EDITING** Left: Distribution of sub-task counts within each complex instruction. Right: Distribution of editing models used during trajectory generation, where Flux.1-Kontext dominates, supplemented by Qwen-Image-Edit, Step1X-Edit, OmniGen2, Bagel, and InstructPix2Pix.

C.3. Reliability of the Termination Mechanism

Table C reports the average number of actual reasoning steps taken by MIRA under different maximum step budgets. As the allowed iterations increase from 3 to 5, the average step count rises moderately (from 2.976 to 4.111). This trend reflects the fact that many instructions in our benchmark consist of multiple sub-tasks, often requiring at least three editing actions to complete. However, when the maximum step budget is further expanded from 5 to 7, the average actual steps remain nearly unchanged (4.096 to 4.208), indicating no meaningful scaling with the larger budget. These results show that MIRA 's termination behavior is goal-driven rather than budget-driven. Specifically:

- It does not exhaust the available step budget, even when more iterations are permitted.
- It prevent unnecessary edits that may introduce extra artifacts.
- It consistently converges to a stable number of reasoning steps on this benchmark ($\sim$ 4 steps on average).

Overall, these findings confirm that the termination controller effectively prevents over-editing and contributes to both the efficiency and robustness of MIRA.

Table C. **Ablation Study on the Reliability of MIRA 's Termination Mechanism.** We report the average number of inference steps taken by MIRA across our 500-sample benchmark when paired with Flux.1-Kontext under different maximum allowed inference step budgets.

Method	Max=3	Max=4	Max=5	Max=6	Max=7
Flux.1-Kontext + MIRA	2.976	3.672	4.111	4.096	4.208

D. Qualitative Analysis

D.1. Excution Error Self-Correction

Figure 5 provides a representative example illustrating MIRA's inherent robustness to execution errors made by the external instruction-guided image editing model. Given the complex instruction: "change the white stove to black, let the floor be wooden, and let the white cabinets be wooden and brown", MIRA decomposes the request into a sequence of visually grounded atomic edits and executes them iteratively.

In the first step (Atomic 1), MIRA successfully guides the base editing model to convert the floor into a wooden texture. However, during Atomic 2, although all the white cabinets are correctly recolored to a wooden and brown appearance, the refrigerator, whose color should remain white, is mistakenly changed to brown. It means that the editing model produces an incorrect or partially incorrect transformation. Because MIRA operates in a closed-loop manner, it does not assume the correctness of any intermediate step. Instead, at each iteration it re-analyzes the current intermediate image I_{t-1} together with the original image I_0 and the input complex instruction C. After executing Atomic 3 correctly, MIRA identifies that the refrigerator still violates the intended appearance and therefore generates a corrective atomic instruction in Atomic 4 ("Change the color of the wooden refrigerator to white"). However, when examin-

ing the Step 4 Output, MIRA identifies a new deviation: the stove, which should remain black, has been unintentionally changed to white during the previous correction step. Then MIRA generates one more corrective atomic instruction in Atomic 5 ("change the color of the white stove to black"). Although this instruction appears repetitive, it reflects an intentional error mitigation strategy that emerges from MIRA 's state-conditioned reasoning. The system then applies this corrective action, and once the resulting image fully satisfies the instruction, MIRA issues the ⟨Stop⟩ signal in Atomic 6.

This case study demonstrates that MIRA can autonomously diagnose discrepancies introduced by the base editing model and dynamically adjust its editing trajectory, effectively mitigating error propagation and preserving alignment with user intent even in the presence of imperfect intermediate transformations.

D.2. More Qualitative Results

Figure C and Figure D show 10 qualitative comparisons for MIRA against leading proprietary and open-source models.

E. User Study

We conduct a user study to further evaluate semantic consistency, perceptual quality, and background preservation across all the editing results. A total of 50 samples were randomly selected from our benchmark, and we recruited 10 participants to assess anonymized outputs independently. For each sample, participants were shown the input image, the editing instruction, and output images from different editing methods presented in a randomized order to minimize positional bias. Ratings were collected on a 0-10 rating scale for all three criteria. This evaluation protocol ensures that the comparison focuses solely on semantic consistency, perceptual quality, and background consistency independent of model identity or architectural differences.

The aggregated scores in Table D reveal clear trends across model categories. Proprietary image editing systems achieve the highest semantic consistency and perceptual quality ratings, reflecting strong capability in precise instruction following and maintaining visual realism. In contrast, models enhanced with MIRA show the strongest performance in background consistency. These results highlight that MIRA substantially improves semantic consistency, perceptual quality, and background preservation for open-source editors.

F. Limitations

MIRA still inherits errors from external editors, relies on synthetic samples, and may struggle with highly ambiguous instructions. Its iterative loop increases latency, and correction steps cannot fully prevent artifact accumulation in complex or visually challenging cases.

Table D. **User Study Results across Semantic Consistency, Perceptual Quality, and Background Preservation.** Participants rated anonymized outputs from proprietary systems, open-source editors, and MIRA-enhanced models on a 0–10 rating scale.

Method	Semantic Consistency ↑	Perceptual Quality ↑	Background Preservation ↑
	Open Source Image Editing Models		
InstructPix2Pix [3]	2.0	5.7	4.3
OmniGen2 [9]	4.6	5.9	6.7
Bagel [4]	5.8	6.0	7.3
Step1X-Edit [6]	6.8	6.2	8.6
Flux.1-Kontext [2]	6.3	6.6	8.6
Qwen-Image-Edit [8]	7.0	7.1	8.7
	Proprietary Image Editing Models		
GPT-Image [7]	8.3	8.9	7.0
Nano-Banana [1]	9.0	8.2	8.9
	Open Source Image Editing Models Plug and Play with MIRA		
Step1X-Edit + MIRA	7.6	6.8	8.6
Flux.1-Kontext + MIRA	7.3	6.8	9.1
Qwen-Image-Edit + MIRA	7.9	7.2	9.0

References

[1] Google AI. Nano banana. 4

[2] Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv e-prints*, pages arXiv–2506, 2025. 4

[3] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18392–18402, 2023. 3, 4

[4] 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. 4

[5] 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. 1

[6] 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. 4

[7] OpenAI. Gpt-image. https://openai.com, 2025. 4

[8] 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. 4

[9] Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, et al. Omnigen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025. 4

[10] Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, and Li Yuan. Imgedit: A unified image editing dataset and benchmark. *arXiv preprint arXiv:2505.20275*, 2025. 1, 2

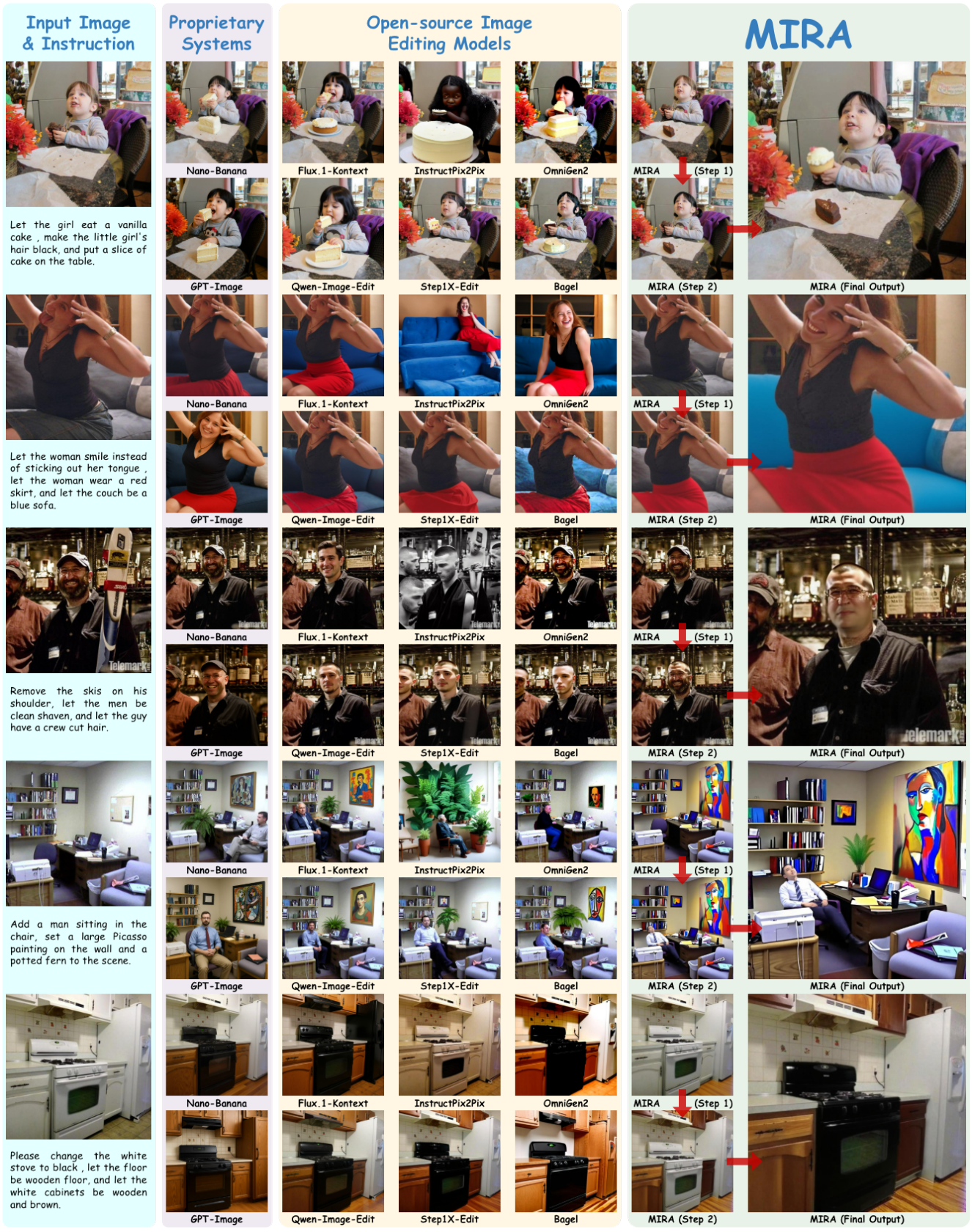

Figure C. Qualitative comparison of MIRA against leading proprietary and open-source image editing models on complex instructions.

Figure D. Qualitative comparison of MIRA against leading proprietary and open-source image editing models on complex instructions.