Amateur-Friendly Conversational Image Editing Agent via Three Stages of Multitask Alignment

Supplementary Material

1. Future Work and Limitations

Our study is limited by (i) a restricted editing space with only 16 global operations, lacking local/semantic controls; (ii) a simplified `matplotlib`-based engine that cannot fully reproduce professional rendering pipelines, creating a sim-to-real gap; (iii) modest data scale and partial reliance on synthetic/pseudo-labeled supervision; (iv) template-driven construction of Image-Refine that may not capture authentic, nuanced user corrections; and (v) evaluation that leans on pixel distances, an internal reward model, and a small user study, which together under-represent subjective aesthetics and long-term satisfaction.

Future work will expand the action space to local and semantic-aware tools (e.g., segmentation-guided or generative-assisted operations) while preserving realism; build a higher-fidelity, Linux-compatible editing backend to reduce latency and deployment mismatch; collect larger and more diverse, human-in-the-loop preference/refinement data; and adopt stronger evaluations via broader, longitudinal user studies and perceptual-quality metrics.

2. List of Notations

We list the notations of symbols used in this paper.

Table 1. Key symbols and their meanings used in the main paper.

Symbol	Meaning
$\mathcal{E}$	Simulation image editor.
Q	User instruction/query text.
t	A single tool call with corresponding parameter.
T	A set of tool parameters (one parameter per tool). $T = \{t_1, \ldots, t_m\}$
$T \setminus \{t\}$	Tool set with tool t removed.
$\mathcal{M}$	IEA which maps multimodal inputs to T (edit/refine) or Q (summary).
I_{ori}	Original unedited image.
I_{ref}	Reference image retouched by human expert.
I_{edit}	Edited image after applying T via the editor $\mathcal{E}$.
I_{his}	User-edited historical image.
$\mathcal{L}(I_a, I_b)$	Pixel-level distance between images (mean of L1 and L2).
L	Shorthand for $\mathcal{L}(I_{\text{edit}}, I_{\text{ref}})$.
R_L	Likeness Improvement Reward.
R_U	Tool Usefulness Reward.
R_A	Summary Alignment Reward.
$\mathbb{1}(\cdot)$	Indicator function (1 if condition holds, else 0).

Table 2. Available image editing tools in our simulation image editor.

Function	Description	Example of Value
exposure	Adjusts the overall image exposure	30 (brighter)
brightness	Adjusts overall image brightness	30 (brighter)
contrast	Adjusts the difference between light and dark areas	40 (higher contrast)
natural_contrast	Adjusts natural contrast	40 (higher contrast)
highlights	Adjusts the brightest areas of the image	-50 (darker highlights)
shadows	Adjusts the darkest areas of the image	50 (lighter shadows)
whites	Adjusts the white point of the image	-20 (duller whites)
blacks	Adjusts the black point of the image	20 (lighter blacks)
saturation	Adjusts the color intensity	-100 (black & white)
vibrance	Boosts muted colors more than saturated colors	50 (more vibrant)
temperature	Adjusts the color temperature (warm/cool)	-20 (cooler)
tint	Adjusts the color tint (green/magenta shift)	50 (more green)
sharpness	Adjusts the clarity of edges	80 (sharper)
vignette	Adds a dark or bright effect to the corners	-30 (darker corners)
fade	Applies a washed-out look to the image	60 (more faded)
grain	Adds film grain or noise to the image	20 (more grain)

3. List of Available Tools

We list all 16 image editing tools used in this paper.

4. Greedy Tool-wise Search Algorithm

We provide a detailed pseudo-code for the algorithm we used during optimal tool search.

Algorithm 1: Greedy Tool-wise Parameter Search.

Input : I_{ori}, I_{ref}, E (editor); initial T (dict; tools $\mapsto$ values in $[-100, 100]$)
Offset set $\Delta = \{\pm 50, \pm 25, \pm 10, \pm 5\}$, threshold $\tau > 0$
Output: Refined tool-call T'

Editing call: $I_{\text{edit}} \leftarrow E(I_{\text{ori}}, T_{\text{edit}})$
Distance: $\mathcal{L}(I_{\text{edit}}, I_{\text{ref}})$
Function `Loss(`T`)`**:**
  $I_{\text{edit}} \leftarrow E(I_{\text{ori}}, T)$
  return $\mathcal{L}(I_{\text{edit}}, I_{\text{ref}})$
end
$L^\star \leftarrow$ `Loss(`T`)`; $S \leftarrow \text{keys}(T)$
while $S \neq \emptyset$ **do**
  $(t^\star, \delta^\star, \text{gain}^\star) \leftarrow (\varnothing, 0, 0)$
  foreach $t \in S$ **do**
    foreach $\delta \in \Delta$ **do**
      $T' \leftarrow T$;
      $T'[t] \leftarrow \text{clip}\big(T[t] + \delta, -100, 100\big)$
      $L' \leftarrow$ `Loss(`T'`)`; gain $\leftarrow L^\star - L'$
      if gain $>$ gain* **then**
        $(t^\star, \delta^\star, \text{gain}^\star) \leftarrow (t, \delta, \text{gain})$
      end
    end
  end
  if gain$^\star \leq \tau$ **then**
    break `// no sufficient improvement`
  end
  $T[t^\star] \leftarrow \text{clip}\big(T[t^\star] + \delta^\star, -100, 100\big)$
  $L^\star \leftarrow L^\star - \text{gain}^\star$; $S \leftarrow S \setminus \{t^\star\}$
end
return T `// T is the refined T'`

5. Ablations on Parameter Search.

We compare six search methods during parameter search process: (1) *InitGen*, the initial attempt from `GPT-4.1`; (2) +*Reflect*, a second search round with reflection; (3) +*SimAnneal*, simulated annealing with 1000 iterations; (4) +*RandomSearch*, random perturbation with $\Delta \in [-100, 100]$ for 8 attempts; (5) +*FastSearch*, our proposed algorithm; and (6) +*Greedy*, exhaustive testing of each parameter from -100 to $+100$. As shown in Figure 1, our search strategy provides a favorable balance between search cost and final quality.

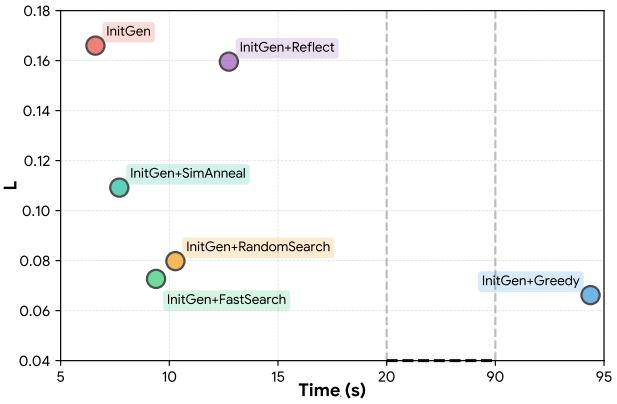

Figure 1. Comparison of parameter search algorithms. Closer to left-bottom is better.

Table 3. Composition of training data across stages. Stage columns are the sampling ratio used.

Image Source	Task	# Items	Sampling Ratio per Stage		
			Stage 1	Stage 2	Stage 3
GIER [6]	Image-Edit	25,763	1	1	2
	Image-Summary	3,583	1	1	5
	Image-Edit-Synthesis	173,802			0.2
	Image-Summary-Synthesis	17,110			1
	Image-Refine-Synthesis	7,166			2
MIT–Adobe FiveK [1]	Image-Edit-Synthesis	174,006			0.2
	Image-Summary-Synthesis	17,102			1
	Image-Refine-Synthesis	7,136			2

6. Data Detail

We provide the detailed training data composition of all three stages.

7. Reward Model Setup

We fine-tune a lightweight judge, `Qwen3-0.6B`[7], to evaluate the consistency between a predicted preference Q_{pred} and a ground-truth preference Q_{ref}. The RM outputs $R_A \in [-10, 10]$ based on semantic alignment, attribute coverage, and specificity. We train with a batch size of 256, learning rate of 5×10^{-5}, for 10 epochs on $\sim$140k SFT items (mixture of synthetic and real), totaling $\sim$5.5k steps. On a 3k-sample test split, we report mean absolute error (MAE) and accuracy as metrics.

Table 4. Results of reward model. MAE $\downarrow$ is absolute error on $[-10, 10]$; Acc $\uparrow$ is the fraction of accurate judgments.

Model	**MAE** ($\downarrow$)	**Acc** ($\uparrow$)
GPT 4.1 mini[4]	5.0013	0.1420
Gemini 2.5 Flash-Lite[2]	5.1923	0.0983
Qwen3-0.6B	7.2412	0.0742
IEA-Summary-RM	1.1381	0.6654

Qualitatively, the SFT RM sharpens discrimination among near-miss summaries (e.g., "brighter but less colorful" vs. "brighter and more colorful"), while penalizing vacuous or off-topic outputs. We visualize predicted vs. ground-truth scores as a 2D histogram in Figure 2. IEA-Summary-RM exhibits tight mass along the diagonal, with heavier tails primarily at extreme scores.

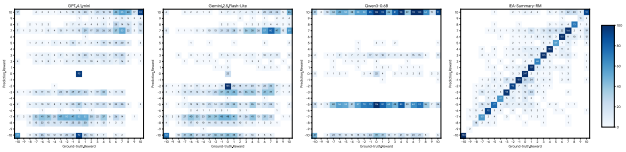

Figure 2. Distribution of predicted reward vs. ground-truth scores.

8. GUI Interface

To better domonstrate the functions of IEA, we build a online demo that supports: **(1)** The user inputs an original image and puts forward requirements: *"Brighten the image to make it clearer."* **(2)** The user provides two groups of historically edited images, and IEA summarizes the style instructions: *"Please, increase color intensity."* **(3)** Based on the previous context, the user puts forward new requirements: *Please increase the yellow tones to give the photo a more vintage appearance.* Users can input commands via natural language, and the IEA will call parameterized tools for editing and present the results.

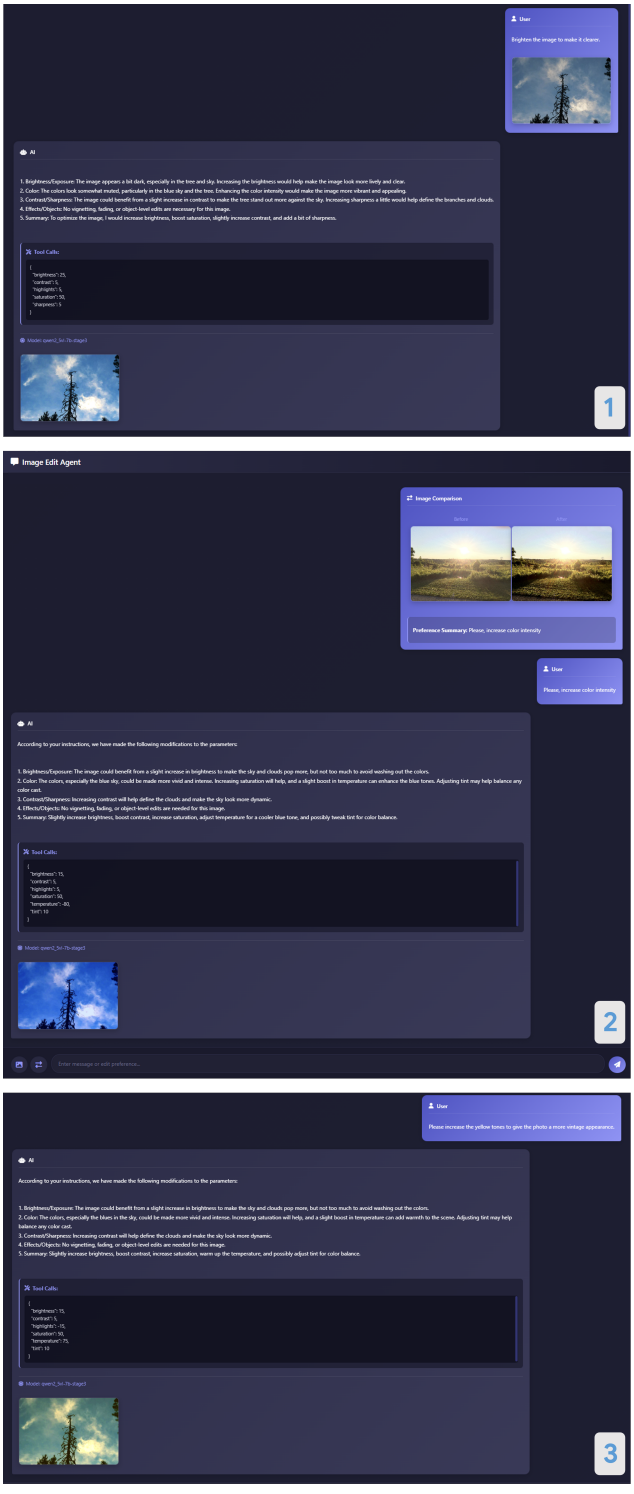

Figure 3. Interface of **I**mage **E**diting **A**gent.

Table 5. Expanded results on the 50 user-study samples. Rank(A/B) are average human rankings for instruction following and image quality, respectively.

Model	Rank (A↓)	Rank (B↓)	L (↓)	R_L (↑)	CLIP (↑)	LLM (Ins↑)	LLM (Sim↑)	LLM (Qua↑)
Reference	2.91	2.87	0.00	1.00	1.00	9.38	9.46	5.72
Origin	-	-	0.17	0.00	0.94	8.94	3.22	4.78
GPT-Image-1	2.93	4.15	0.22	-0.34	0.91	9.60	8.78	5.50
Qwen-Image-Edit	3.26	3.74	0.18	-0.13	0.91	9.58	7.54	5.76
PatchDPO	-	-	0.27	-0.60	0.85	8.96	4.36	5.18
GenArtist	-	-	0.19	-0.16	0.92	9.15	4.54	4.96
JarvisArt	6.13	5.36	0.22	-0.33	0.91	9.56	4.78	5.70
GPT-4.1	5.86	5.84	0.21	-0.31	0.91	8.24	4.80	5.22
Gemini-2.5-Pro	5.47	5.79	0.23	-0.42	0.88	7.62	4.74	5.32
Qwen2.5-VL-7B	4.80	4.55	0.20	-0.22	0.93	8.30	5.62	5.12
Ours	4.64	3.69	0.13	0.14	0.94	9.40	6.18	5.62

9. User Study Detail

Figure 4 shows a screenshot of the ranking interface used in our user study, illustrating how participants evaluated the different editing outputs for a given instruction.

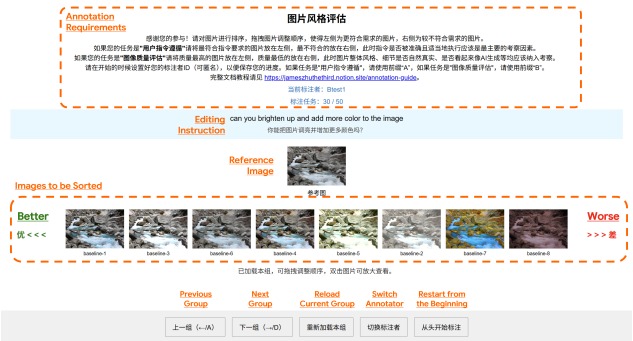

Figure 4. Screenshot of the user study interface. For a given original image and instruction (top), participants were asked to rank the edited results from different methods (bottom) by dragging and dropping them in order of preference or compliance.

We report Friedman tests and Kendall's W for inter-user agreement in the user study(Table 6). Both tasks show high significance ($p<10^{-7}$). Pairwise comparisons confirm $p<0.05$ for all baseline pairs adjacent ranks in Task A, and all pairs except Gemini(7th) vs. GPT-4.1(8th) in Task B. The moderate global W reflects natural variation in aesthetic preferences, while per-image $W\approx0.5$ indicates **reasonable agreement** when judging individual samples.

Table 6. Statistical significance of user study rankings.

Task	$\chi^2(7)$	p-value	Kendall's W	Per-image W
A: Instruction Following	5544.46	$< 10^{-7}$	0.288	0.505 ± 0.162
B: Image Quality	3832.96	$< 10^{-7}$	0.196	0.544 ± 0.169

10. Additional User Study Results

Besides the compact user-study table in the main paper, we additionally report expanded automatic and judge-based metrics on the same 50 evaluation samples in Table 5. We include two more generative baselines: `PatchDPO` [3] and GenArtist [8]. Besides L1/L2 distance metrics used in main paper, we also include Cosine Similarity with reference image using features extracted by CLIP [5], and also employ LLM-as-Judge (gemini-2.5-pro) to rate 0-10 score based on **Ins**truction Following, Reference **Sim**ilarity, and Image **Qua**lity. IEA shows **consistant and competitive performance** among vaiours metrics and baselines.

11. Prompts Used

System Prompt for Image Analysis

You are an expert image editor. You know how to edit a photo and make it look nicer. In this task, you will be given an input image, a reference output image, and a text instruction that describes the task. Your task is to determine the JSON object of parameters that were used in an image editor to transform the input image into the output image.

The available parameters are described in the following table:
{image_edit_tools}

User Prompt for Image Analysis

```
{instruction_template}

{operator_constraint}

{chain_of_thought}
<answer>
```json
{
    "exposure": 10,
    "sharpness": -30,
    ... // other parameters
}
```
</answer>
```

Instruction templates of User Prompt for Image Analysis

```
-1: "The user instruction is not
    provided. You need compare and
    analyze the input and output
    images to determine the
    adjustments made.",
0: "You can follow user instructions
    to determine the adjustments made
    :\n(From User) {instruction}",
1: "You can follow expert instructions
     to determine the adjustments made
    :\n(From Expert) {instruction}",
2: "You can follow user preferences to
     determine the adjustments made:\n
    {instruction}"
```

Operator constraint templates of User Prompt for Image Analysis

```
0: "If a function or parameter is not
    mentioned in the instruction, it
    could also be included in the
    output. Analyze the images and
    instruction to determine which
    functions and parameters are
    relevant.",
1: "You MUST include and ONLY use the
    following operators in your output
    : \n{operators}\nDo not include
    any other operators.",
2: "During previous attempts, You are
    told that you MUST include and
    ONLY use selected functions. If
    you are sure that the used
    functions are not enough to make
    the image nicer, you can ignore
    the operator constraint and use
    any available functions provided
    above in the table."
```

Chain of thought templates of User Prompt for Image Analysis

```
0: "Your output should be a dictionary
     in JSON format with the modified
    parameters:",
1: "Think step by step to determine
    the adjustments made in the image
    edit. Consider the input and
    output images, and the user
    instruction if provided. First,
    provide your thoughts about how to
     optimize the input image,
    reasoning just like your usual
    tasks that ONLY an input image is
    provided. DO NOT reveal that you
    have known the user instructions
    or the reference output image (DO
    NOT mention them!). DO NOT contain
     exact parameter numbers but use
    expressions of scale. You should
    follow the template below:\n<think
    > your thoughts about how to make
    this photo nicer </think>",
2: """Think step by step to determine
    the adjustments made in the image
    edit. Consider the input and
    output images, and the user
    instruction if provided. First,
    provide your thoughts about how to
```

```
     optimize the input image,
    reasoning just like your usual
    tasks that ONLY an input image is
    provided. DO NOT reveal that you
    have known the user instructions
    or the reference output image (DO
    NOT mention them!). DO NOT contain
     exact parameter numbers but use
    expressions of scale. You should
    follow the template below:
<think>
1. Brightness/Exposure: Check if the
    image needs to be brighter or
    darker overall (use brightness,
    exposure, highlights, shadows,
    whites, blacks).
2. Color: Note any possible change in
    color intensity or warmth (use
    saturation, vibrance, temperature,
     tint).
3. Contrast/Sharpness: See if contrast
     or sharpness should be changed (
    use contrast, sharpness, fade,
    grain).
4. Effects/Objects: Look for possible
    vignetting, fading, cropping,
    rotation, or object-level edits (
    crop, rotate, flip, inpaint_obj,
    rotate_obj, etc).
5. Summary: List all likely parameter
    changes and values that can make
    the input image look nicer.
</think>
""",
    3: """Think step by step to
        determine the adjustments made
        :
<think>
1. General Optimization:
   - Brightness/Exposure: Check if the
        image needs to be brighter or
       darker
   - Color: Note any color adjustments
        needed
   - Contrast/Sharpness: Evaluate
       contrast and sharpness
   - Effects/Objects: Identify needed
       edits like cropping or rotation

2. User Preference Analysis:
   {preference_analysis}

3. Integrated Adjustments: Combine
    general improvements with user
    preferences
</think>
"""
```

User Prompt for Image Reflection

```
{instruction_template}
{operator_constraint}

Parameters used in the previous best
round of edit:
{best_params}
```

You should reflect on the previous best edit and determine if any changes to parameters or functions are needed.

```
{chain_of_thought}
<answer>
```json
{
    "exposure": 10,
    "sharpness": -30,
    ... // other parameters
}
```
</answer>
```

Chain of thought templates of User Prompt for Image Reflection

```
0: "Your output should be a dictionary
     in JSON format with the modified
    parameters:",
1: "Think step by step to determine
    the adjustments made in the image
    edit. Consider the input and
    output images, and the user
    instruction if provided. First,
    provide your thoughts about how to
     optimize the input image,
    reasoning just like your usual
    tasks that ONLY an input image is
    provided. DO NOT reveal that you
    have known the user instructions
    or the reference output image or
    the previous edit image (DO NOT
    mention them!). DO NOT contain
    exact parameter numbers but use
    expressions of scale. You should
    follow the template below:\n<think
    > your thoughts about how to make
    this photo nicer </think>",
```

```
2: """Think step by step to determine
    the adjustments made in the image
    edit. Consider the input and
    output images, and the user
    instruction if provided. First,
    provide your thoughts about how to
     optimize the input image,
    reasoning just like your usual
    tasks that ONLY an input image is
    provided. DO NOT reveal that you
    have known the user instructions
    or the reference output image or
    the previous edit image (DO NOT
    mention them!). DO NOT contain
    exact parameter numbers but use
    expressions of scale. You should
    follow the template below:
<think>
1. Brightness/Exposure: Check if the
    image needs to be brighter or
    darker overall (use brightness,
    exposure, highlights, shadows,
    whites, blacks).
2. Color: Note any possible change in
    color intensity or warmth (use
    saturation, vibrance, temperature,
     tint).
3. Contrast/Sharpness: See if contrast
     or sharpness should be changed (
    use contrast, sharpness, fade,
    grain).
4. Effects/Objects: Look for possible
    vignetting, fading, cropping,
    rotation, or object-level edits (
    crop, rotate, flip, inpaint_obj,
    rotate_obj, etc).
5. Summary: List all likely parameter
    changes and values that can make
    the input image look nicer.
</think>
"""
```

System Prompt for Summary Reward

You are an expert image editor. You know every function and parameter as well as their effect in an image editor. You will be given a user instruction and a summarized user preference. Your task is to determine whether the summarized user preference is consistent with the user instruction and report a consistent score between -10 and 10. You can follow the cases below to determine the score:
Reference: Make the image brighter and vivid, and add sharpness to it.
Prediction 1: The user prefers a brighter and more colorful image, also make the image sharper.
Score 1: 10
Prediction 2: The user prefers a brighter and less colorful image.
Score 2: -5
Prediction 3: The user prefers a image high contrast and sharpness.
Score 3: 3
Prediction 4: The user wants to make the image nicer.
Score 4: 0
Prediction 5: Today is a sunny day. (Irrelevant answer, or in a different language, or kept repeating the same sentence, or other nonsense)
Score 5: -10

User Prompt for Summary Reward

Now, give your score as a single integer between -10 and 10 based on the user instruction and the summarized user preference; do not output any other text.

```
Reference: {reference}
Prediction: {prediction}
Score:
```

System Prompt for Generating Instructions

You are an expert image editor. You know every function and parameter as well as their effect in an image editor. You will be given a group of slider parameters (range -100 100) that were used to edit a photo. Your task is to convert them into a concise, natural, and specific image editing instruction, as if it were written by an amateur user.

The available parameters are described in the following table:
`{image_edit_tools}`

User Prompt for Generating Instructions

Given the following image editing parameters and related technical instruction written by an expert, generate a concise and natural image editing instruction that an amateur user might give (i.e., I want.../Make it.../Could you.../This image is too...). Do not mention numeric values or technical terms. Focus on describing the visual effect or style preferred in everyday language, using adverbs

of degree where appropriate, instead of listing parameters. Do not output any other text.

```
Parameters:
{params}
Expert Instruction:
{expert_instruction}
Amateur User Instruction:
```

System Prompt for Intent Following Judger

You are an image editing evaluator. Evaluate how well the edited image, produced by modifying the original according to the given instruction, adheres to the specified editing requirements. You are only required to judge how closely the edit follows the instruction instead of the image quality. Score from 0 to 10:
* 10: Perfectly follows instruction, i.e., all changes are appropriate
* 7 - 9: Mostly follows with minor issues, i.e., missing minor changes, the modifications are over/under done
* 4 - 6: Partially follows but significant issues, i.e., some changes are inappropriate or missing major changes, or just like the original image
* 1 - 3: Makes opposite changes, i.e., changes are irrelevant and contradict the instruction with negative impact, the image looks obviously worse
* 0: Completely wrong, i.e., the image is entirely changed, unclear, or distorted in a way that contradicts the instruction

Please focus on the detail shift between two images, the change could be small but important.
Return only the score as an integer between 0 and 10.

User Prompt for Intent Following Judger

```
**Original Image:**
<original_image>
**Edited Image:**
<edited_image>
**Editing Instruction:**
{instruction}
```

Score:

System Prompt for Target Gap Judger

You are an image editing evaluator. Evaluate how closely the expert-provided target image aligns with the image produced by editing the original image according to the given instruction. You are not required to judge how closely the edit follows the instruction. Instead, consider the visual similarity between edited image and target image. Score from 0 to 10:
* 10: Edited image matches target perfectly, i.e., the edited image is nearly identical to the target image
* 7 - 9: Close to target with small differences, i.e., minor shift on colors or brightness, the edited image looks more similar to target than original image
* 4 - 6: Some similarity but major differences, i.e., significant color/brightness changes, blurs, or artifacts, or just like the original image
* 1 - 3: Completely different from target, i.e., wrong objects, extreme distortions, or irrelevant edits, not similar to either target or original image
* 0: No resemblance to target, i.e., the image is entirely changed, unclear, mosaic, or distorted in a way that contradicts the target

Please focus on the similarity between images, if the edited image is very close to the target image, give a high score, if the edited image is very close to the original image, give a near zero score, if the edited image is very different from the target image and the original image, give a negative score. Please focus on the detail shift between three images, the change could be small but important.
Return only the score as an integer between 0 and 10.

User Prompt for Target Gap Judger

```
**Original Image:**
<original_image>
**Edited Image:**
<edited_image>
**Target Image:**
<target_image>
**Editing Instruction:**
{instruction}
```

Score:

System Prompt for Image Quality Judger

You are an image editing evaluator. Evaluate the quality of the image produced by editing the original image according to the given instruction. You are not required to judge how closely the edit follows the instruction. Instead, consider the overall visual style, the naturalness and realism of the details, and any indications that the image may be AI-generated. Score from 0 to 10:
* 10 : Edited image is of high quality, i.e., clear, realistic, well-exposed, etc.
* 7 - 9: Edited image is generally good with minor flaws, i.e., slight noise, imperfect exposure, etc.
* 4 - 6: Edited image has noticeable quality issues, i.e., low resolution, unnatural details, weird colors, etc.
* 1 - 3: Edited image is of poor quality, i.e., blurry, artifacts, unrealistic, mosaic, etc.
* 0: Edited image is of very low quality, i.e., pure white/black, severely distorted, extremely blurry, or completely unrealistic
Please focus on the quality of the edited image itself, both general appearance and fine details.
Return only the score as an integer between 0 and 10.

User Prompt for Image Quality Judger

```
**Original Image:**
<original_image>
**Edited Image:**
<edited_image>
**Editing Instruction:**
{instruction}
```

Score:

References

[1] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. Learning photographic global tonal adjustment with a database of input / output image pairs. In *The Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 4

[2] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 5

[3] Qihan Huang, Long Chan, Jinlong Liu, Wanggui He, Hao Jiang, Mingli Song, and Jie Song. PatchDPO: Patch-level dpo for finetuning-free personalized image generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18369–18378, 2025. 6

[4] OpenAI. Introducing gpt-4.1 in the api. https://openai.com/index/gpt-4-1/, 2025. 5

[5] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 6

[6] Jing Shi, Ning Xu, Yihang Xu, Trung Bui, Franck Dernoncourt, and Chenliang Xu. Learning by planning: Language-guided global image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13590–13599, 2021. 4

[7] Qwen Team. Qwen3 technical report, 2025. 5

[8] Zhenyu Wang, Aoxue Li, Zhenguo Li, and Xihui Liu. GenArtist: Multimodal llm as an agent for unified image generation and editing. *Advances in Neural Information Processing Systems*, 37:128374–128395, 2024. 6