RetouchAgent: Towards Interactive and Explainable Image Retouching with MLLM Agents

Shuo Zhang, Xinyu Yang*

Xi'an Jiaotong University, Xi'an, China

Abstract

Although deep learning-based image retouching has made significant progress, its inherent subjectivity renders current black-box methods limited in interactivity and explainability. Among existing efforts, parameter-controlled methods aim to improve interactivity, but often suffer from ambiguous semantics and lack support for natural language control. Reinforcement learning–based explainability methods are constrained by low-dimensional and limited action spaces, which result in suboptimal performance. To address the above issues, we propose RetouchAgent, a novel framework that leverages collaboration among multiple MLLM agents for image retouching. Our method consists of the following key steps: (1) Retrieval: By constructing a multimodal retouching database, we enable an ICL sample retrieval mechanism guided by retouching intent. (2) Engine: Leveraging the vision-language understanding capabilities of MLLM, a carefully designed prompting strategy, and a dedicated operation library, we enable precise and controllable image retouching. (3) Reflection: We evaluate each retouching interaction and optimize the retouching process for progressive result refinement. Finally, through multiple rounds of collaboration among MLLM agents, RetouchAgent achieves state-of-the-art performance in quantitative and qualitative evaluations.

Code — https://github.com/Cooperxjt/RetouchAgent

Introduction

Image retouching is crucial in modern image processing. Users often adjust attributes like exposure and color to achieve more desirable images. While professional software like Photoshop offers powerful tools, their complexity can be overwhelming for non-experts. To address this, automated image retouching algorithms have emerged, particularly black-box techniques, such as those based on three-dimensional lookup table (3D LUT), have achieved significant progress (Zeng et al. 2022; Yang et al. 2022; Zhang et al. 2022; Li et al. 2024b,a; Yang et al. 2025). However, given the subjective nature of the task, these methods remain limited in both interactivity and explainability.

Interactivity demands methods that can respond effectively to user intent by generating diverse retouching strategies, even when the input is ambiguous. Some methods incorporate controllable parameters within the network to address this issue (Song, Qian, and Du 2021; Kim and Lee 2024; Kim, Koh, and Kim 2020; Duan et al. 2025). However, the unclear link between the parameters and the results requires users to learn, which limits practical usability. Explainability means the methods should provide operations that are transparent and interpretable, helping users understand and guide the retouching process. Reinforcement learning–based methods (Park et al. 2018; Hu et al. 2018; Kosugi and Yamasaki 2020; Ke et al. 2022; Gao et al. 2024; Zhang et al. 2024) attempt this by generating action sequences. However, the action space is usually limited to a few low-dimensional controls like brightness and contrast, which makes it hard to handle visual details and affects the final output quality.

Recently, Multimodal Large Language Model (MLLM)-based autonomous AI agents have demonstrated effectiveness in handling complex tasks (Gu et al. 2024; Agarwal et al. 2024; Xue et al. 2025; Li et al. 2025). Equipped with strong visual-language understanding capabilities, multiple agents can collaboratively execute user instructions, even under open-world conditions. Inspired by this, we present RetouchAgent, a novel framework that leverages multi-agent MLLM collaboration to perform interactive and explainable image retouching.

To achieve this, several key challenges must be addressed. First, adapting MLLMs to artistic tasks remains challenging. Although they demonstrate strong In-Context Learning (ICL) abilities, enabling inference from a few examples without parameter updates, standard visual ICL methods rely solely on semantic similarity for example retrieval (Brown et al. 2020; Dong et al. 2024; Zhou et al. 2024a). This strategy may fail in practice, as a single image can have multiple valid retouching styles. To meet diverse user needs, intent-aware ICL examples retrieval is essential. Moreover, while MLLMs have been applied to direct image editing, this can compromise authenticity by altering image content, as shown in Fig. 1. A safer solution is to use public editing libraries like GMIC, but their complex interfaces remain difficult for MLLMs to interpret and control. Finally, iterative refinement is crucial in practical image retouching, yet the

*Corresponding author.

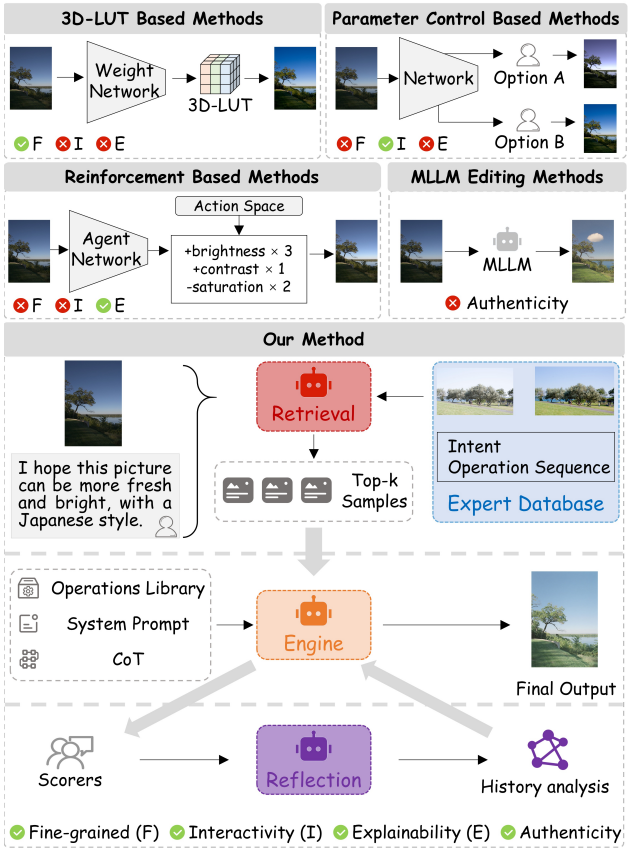

Figure 1: Comparison of different methods. By coordinating multiple MLLM agents, our approach combines fine-grained retouching flexibility, seamless natural language interaction, enhanced explainability, and authenticity.

absence of structured guidance makes it difficult for single-step reasoning to match the results of expert workflows.

To address these challenges, our RetouchAgent consists of three components. We first design a retrieval agent supported by a dedicated data construction pipeline. By extending existing datasets and incorporating user intent, our system can retrieve examples to form intent-aware ICL prompts. Next, our retouching agent uses an operation library with simple, single-parameter controls that support a wide range of retouching functions. By leveraging designed Chain-of-Thought (CoT) prompts, the retouching agent performs image retouching using only lightweight operations without causing image distortion. Finally, we incorporate a reflection agent that emulates expert retouching workflows through iterative analysis of the image retouching process, leading to improved retouching quality.

Our main contributions can be summarized as follows:

- We propose RetouchAgent, an image retouching framework that leverages MLLMs to enable more interactive and explainable retouching workflows.
- By combining multimodal intent-aware ICL retrieval, a lightweight and efficient operation library, and the reflection mechanism, our approach enables MLLM retouching with both high quality and content fidelity.
- Extensive experiments show that RetouchAgent consistently outperforms existing state-of-the-art methods in both quantitative retouching evaluations and real-world user scenarios.

Related Work

Image Retouching

With the release of the MIT-Adobe FiveK dataset (Yan et al. 2016), deep learning-based image retouching has become a major research focus. Among these, 3D LUT techniques have gained attention for their efficiency in color mapping (Zeng et al. 2022; Zhang et al. 2022; Yang et al. 2025). While they offer strong pixel-level performance, they are still limited in interactivity and explainability.

For interactivity, some methods adopt parameter-based controls to adjust retouching styles, such as PieNet (Kim, Koh, and Kim 2020), StarEnhancer (Song, Qian, and Du 2021), and DiffRetouch (Duan et al. 2025). However, the parameters are often implicit and non-intuitive, resulting in limited user accessibility. To improve explainability, other studies have explored reinforcement learning by simulating expert action sequences. DandR (Park et al. 2018) proposes a deep reinforcement learning approach based on a predefined set of operations, while RSFNet (Ouyang et al. 2023) introduces a white-box framework using parallel region-specific filters. Nonetheless, constrained by the limited expressiveness of interpretable operations, such methods often underperform compared to pixel-level approaches. Furthermore, generative models have been introduced into image retouching (Cao et al. 2023; Brooks, Holynski, and Efros 2023; Wang et al. 2024b). But their inherently destructive operations risk compromising the original content, raising concerns about authenticity and identity preservation.

Multimodal Large Language Model

Built on large language models (OpenAI et al. 2024a; Touvron et al. 2023), MLLMs extend input modalities to images, audio, and video, laying the basis for embodied AI. BLIP-2 (Li et al. 2023) introduces a Q-Former for cross-modal query extraction, while Qwen-VL (Bai et al. 2023) integrates a bilingual language model for cross-lingual multimodal tasks. MLLMs support diverse applications: RetouchGPT (Xue et al. 2025) enables interactive face retouching via defect repair and user guidance. AnomalyGPT (Gu et al. 2024)enhances semantic alignment for industrial anomaly detection. Inspired by these advances, our RetouchAgent leverages MLLMs' vision-language capabilities to model the relationship between input images and expert retouching examples, enhancing both interactivity and explainability.

In-Context Learning

In-Context Learning enables MLLMs to perform tasks by conditioning on a few input-output examples along with the test input, without updating model parameters and has been applied across various domains (Lee et al. 2025; Zhou et al.

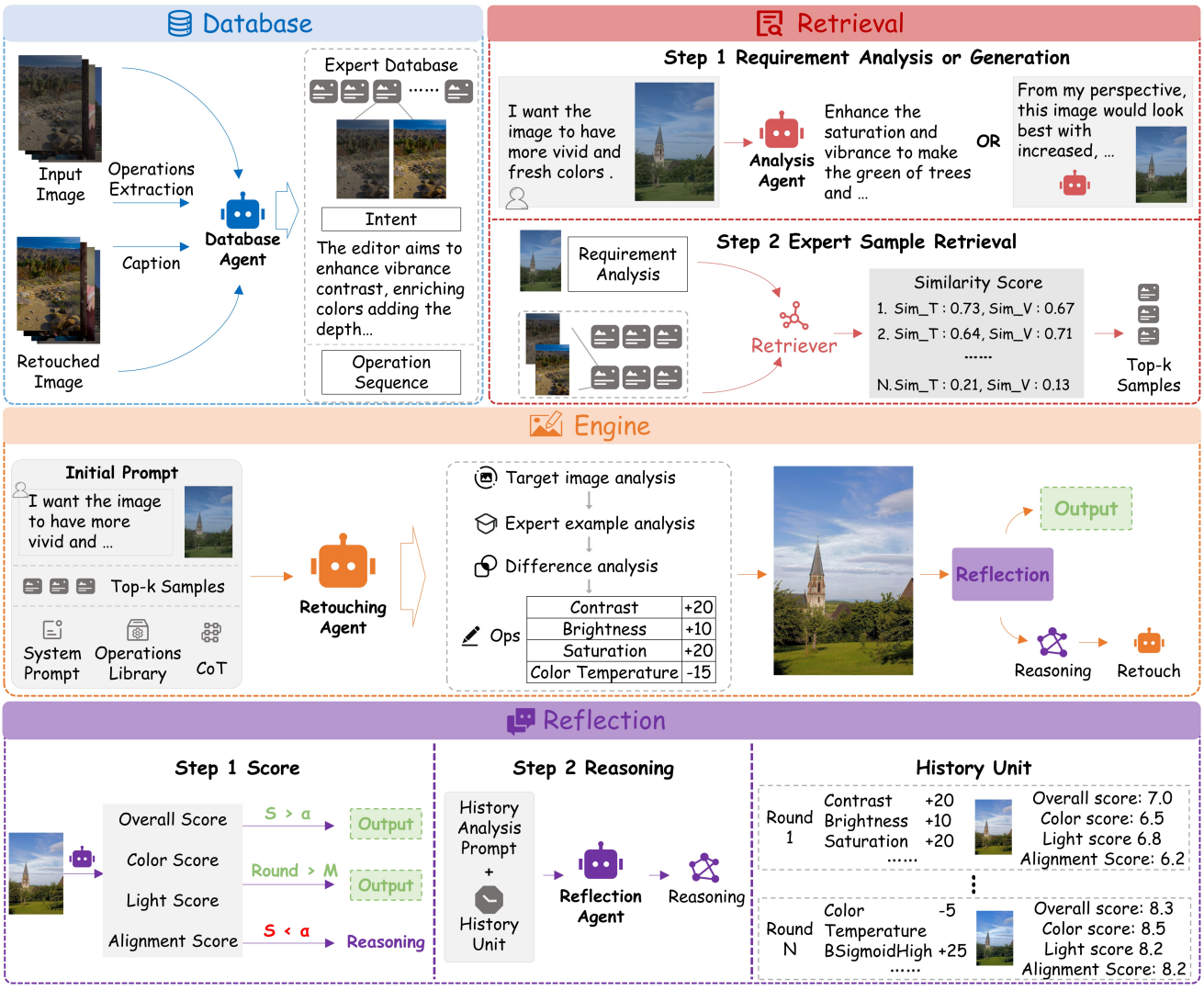

Figure 2: The overview of RetouchAgent. The figure illustrates the architecture of our method, which comprises four main components: Database, Retrieval, Engine, and Reflection.

2024b; Wang et al. 2023). Existing methods typically retrieve semantically similar images to build ICL prompts, but this is limited in image retouching scenarios, where a single image may correspond to multiple styles and user intents (Song, Qian, and Du 2021). To address it, RetouchAgent introduces a multimodal retrieval strategy guided by retouching intent, jointly considering image content and user intent during prompt construction.

Methods

The overall architecture of RetouchAgent is illustrated in Fig. 2. After obtaining the expert database aligned with retouching intent, the retrieval agent leverages the input image and user requirements to retrieve appropriate ICL samples, constructing a prompt that is forwarded to the retouching engine. The retouching agent then selects suitable operations from our designed operation library and generates parameters to retouch. Finally, the outputs are evaluated by scorers, and refined iteratively through a reflection mechanism to produce the final output.

Database Generation

To support intent-aware ICL in retouching tasks, we develop a pipeline that transforms any existing retouching datasets into a structured multimodal expert database, as shown in Fig. 2.

For the retouching dataset $D = \{(i_n, i_n^{re})\}_{n=1}^{N}$ with N images, where i and i^{re} denote images before and after retouching. We first apply a caption model to produce descriptions, aiming to better guide fine-grained retouching operations such as highlights, shadows, and other local details. Subsequently, we extract the expert retouching operation sequence q for each image and remove non-informative operations (e.g., import/export commands). Combining these

operation sequences with paired images, we build a prompt to guide the database agent to infer the reasons behind expert retouching, denoted as p. In our study, we further invite five professional retouchers to evaluate the correctness of the results. As a result, we get a new multimodal database $D = \{(i_n, i_n^{re}, q_n, p_n)\}_{n=1}^{N}$ for subsequent retrieval.

ICL Sample Retrieval

Considering that MLLMs lack fine-tuning for retouching tasks, injecting external expert knowledge via ICL examples offers a practical and flexible alternative. However, the inherent subjectivity of image retouching limits the effectiveness of traditional semantic retrieval paradigms. To overcome this limitation, we propose a two-stage retrieval agent, as shown in Fig. 2, which jointly considers both image content and inferred user intent.

In Step 1, for the input image I and user instruction R, we first employ the analysis agent A_{ana} to parse them. As user instructions are often vague or under-specified, A_{ana} combines visual and textual cues to infer a concrete retouching objective K. If R is omitted, a generic retouching goal is generated based solely on the image. This process is formalized as $K = A_{\text{ana}}(I, R)$ or $K = A_{\text{ana}}(I)$.

In Step 2, the retrieval agent computes the similarity score between the input pair (I, K) and each sample in the expert database $D = \{(i_n, i_n^{re}, q_n, p_n)\}_{n=1}^{N}$, using the following formulation:

$$s_n = \lambda_{vis}\, S_{\text{img}}\big(I, i_n\big) + \lambda_{text}\, S_{\text{text}}\big(K, p_n\big), \qquad (1)$$

where λ_{vis} and λ_{text} are the weights, and S_{img} and S_{text} denote image and text similarity functions, respectively. In our implementation, we adopt pre-trained CLIP (Radford et al. 2021) for feature extraction and compute cosine similarity for both modalities. After ranking all samples by s_n, we select the top-k most relevant ones to form the final ICL prompt output.

Retouching Engine

After retrieving the ICL samples, the retouching engine performs the image retouching process, as shown in Fig. 2. The engine takes the system prompt as input and selects operations from our designed library via hierarchical CoT guidance. Each part is detailed in the sections below.

System Prompt The system prompt defines the role of the MLLM agent, specifies the structure and semantics of the input image and expert examples, and incorporates reference exemplars to enforce standardized formatting.

Operations Library Existing image enhancement open libraries, such as GMIC, OpenCV, and Pillow, either expose overly complex interfaces that are difficult for MLLMs to interpret and control, or they lack the expressive power required for high-quality image retouching. To ensure both interpretability and expressive capability, we design a set of image retouching operations inspired by professional tools such as Lightroom and grounded in expert retouching practices.

The operation library provides a representative subset of professional editing platforms, covering both global image attributes (e.g., exposure, color temperature, contrast) and local refinements (e.g., tone curves, highlights, and shadows). Each operation is controlled using only a single normalized value within the range $[-100, 100]$, which reduces parameter complexity and facilitates continuous control. This design lowers the reasoning load for MLLMs, allowing the retouching agent to focus on image understanding rather than code generation. Finally, the model outputs a structured JSON file that encodes the full plan, enabling an interpretable, consistent, and non-destructive retouching workflow.

Chain-Of-Thought The Chain-Of-Thought process comprises four stages: target image analysis, expert example analysis, difference and historical analysis, and retouching operation generation.

- Target image analysis: The agent first analyzes the input image along with the user's requirements to define actionable retouching objectives. By transforming ambiguous or subjective descriptions into concrete retouching directives and identifying key regions and visual attributes, this step establishes a foundation for subsequent reasoning.
- Expert example analysis: Based on the retrieved ICL examples, the agent performs a detailed analysis to understand the retouching strategies demonstrated by experts. This includes recognizing stylistic patterns, parameter preferences, and localized adjustments illustrated in the examples.
- Difference and historical analysis: Within the initial prompt, the MLLM is guided to compare the target image with expert references to identify priority adjustments. In later reflection iterations, the agent incorporates historical context and feedback from the reflection agent to refine its reasoning and adjust the optimization trajectory accordingly.
- Retouching operation generation: Informed by the reasoning steps above, the retouching agent selects appropriate operations from our designed operation library and generates corresponding parameters.

Upon completing a retouching iteration, the output is forwarded to the reflection agent for further analysis.

Reflection

Even for professional retouchers, image retouching is rarely a one-step operation. Instead, it is an iterative process involving continuous evaluation, refinement, and optimization. Motivated by this observation, we incorporate a reflection agent into RetouchAgent, as illustrated in the Fig. 2.

In Step 1, we first employ the MLLM to evaluate the retouched image on a 10-point scale across four dimensions: overall aesthetics, lighting quality, color fidelity, and intent alignment. The first three criteria focus on visual quality, while the last evaluates whether the result aligns with the user's intended style. If all scores exceed predefined thresholds, the retouched image is accepted as the final output. Otherwise, it proceeds to Step 2 for further refinement.

Method	ExpertA		ExpertB		ExpertC		ExpertD		ExpertE		Average	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
CSRNet	21.37	0.883	22.88	0.895	24.12	0.894	20.09	0.817	21.85	0.838	22.06	0.865
3D-LUT	21.98	0.887	24.39	0.926	25.12	0.917	22.61	0.902	23.19	0.910	23.46	0.908
DandR	20.49	0.855	22.11	0.889	22.15	0.853	20.91	0.850	21.97	0.863	21.53	0.862
RSFNet	22.05	0.891	23.41	0.935	23.02	0.901	23.24	0.913	22.68	0.902	22.88	0.908
PIENet	21.42	0.879	25.74	0.943	24.76	0.912	22.49	0.897	23.93	0.923	23.67	0.911
StarEhancer	21.13	0.876	25.38	0.940	25.09	0.917	23.32	0.915	23.91	0.913	23.77	0.912
TSFlow	20.81	0.871	24.89	0.948	25.44	0.931	22.16	0.906	23.35	0.930	23.33	0.917
DiffRetouch	22.34	0.901	25.82	0.953	**26.13**	**0.940**	24.19	**0.946**	24.05	0.944	24.51	0.937
Ours	**23.13**	**0.913**	**25.85**	**0.955**	26.02	0.938	**24.34**	0.944	**24.13**	**0.950**	**24.69**	**0.940**

Table 1: Quantitative comparison on the MIT-Adobe FiveK dataset with subsets retouched by five experts (A/B/C/D/E). All models are reimplemented by us. The best result is in **bold**, whereas the second-best is underlined. The experiments are done on the 480p setting.

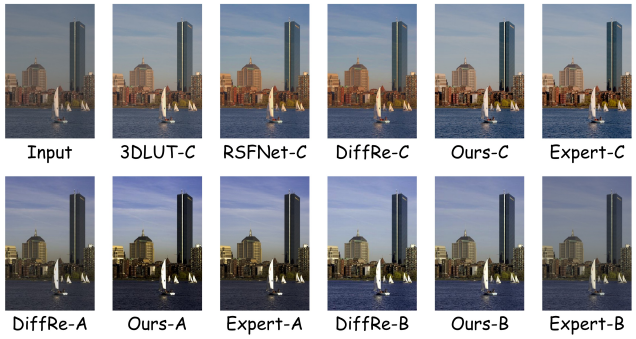

Figure 3: Qualitative comparison on the MIT-Adobe FiveK dataset.

Method	ExpertA	ExpertB	ExpertC
HDRNet	23.01	23.17	23.34
CSRNet	23.86	23.70	23.87
RSFNet	25.18	24.81	25.32
STAREnhacher	25.38	24.92	25.47
3D-LUT	25.63	25.16	25.24
DiffRetouch	25.98	25.45	25.53
Ours	**26.12**	**25.52**	**25.70**

Table 2: Quantitative comparison on the PPR10K dataset with subsets retouched by three experts. All models are reimplemented by us. The best result is in **bold**, whereas the second-best is underlined. The experiments are done on the 340p setting.

During the Step 2, the reflection agent analyzes the sequence of previous operations to identify potential causes of suboptimal results. To support the process, we introduce a history unit that stores both the operation sequences and the outputs from each iteration. Leveraging this structured memory, the agent performs targeted diagnostics and generates actionable guidance for the next round of retouching. It is worth noting that, instead of restarting from scratch, the agent follows an accumulative optimization strategy, progressively refining the image in line with professional retouching workflows. Experimental results demonstrate that this mechanism consistently improves image quality.

Experiments

Datasets and Evaluation Metrics

We evaluate our method on the MIT-Adobe FiveK and PPR10K datasets. The MIT-Adobe FiveK dataset (Yan et al. 2016) contains 5,000 images, which are retouched by five retouching experts (A/B/C/D/E). We randomly select 500 images for testing, while the remaining images are used to construct the expert database. Notably, we compare the results from all five experts to demonstrate the efficient learning capability of our method from expert samples. All the images are resized to 480p in experiments. The PPR10K dataset (Liang et al. 2021) contains 11,161 portrait images, each retouched by three experts. Following (Liang, Zeng, and Zhang 2021), we select 2,286 images for testing, while the other images are used to construct the expert database. Experiments are conducted on the 360p version as commonly used. We adopt PSNR and SSIM to measure pixel-level fidelity and perceptual quality, respectively.

Implementation Details

We select GPT-4o (OpenAI et al. 2024b) to serve as all MLLM agents. In the retrieval module, we employ the pre-trained ViT-B/32 CLIP model as the visual and textual feature extractor. We set the weight parameters λ_{vis} to 0.6, λ_{text} to 0.4, and the number of ICL samples k to 3. In the reflection stage, we set the score threshold to 8.0 and the number of maximum iterations at 8.

Comparison with Other Methods

We conduct the experiments on the five expert annotations of MIT-Adobe FiveK dataset. We compare our method with three kinds of baselines. Single-style methods, including CSRNet (He et al. 2020) and 3D-LUT (Zeng et al. 2022), are reimplemented on the five expert annotations. White-box methods, including Distort-and-Recover (Park et al. 2018)

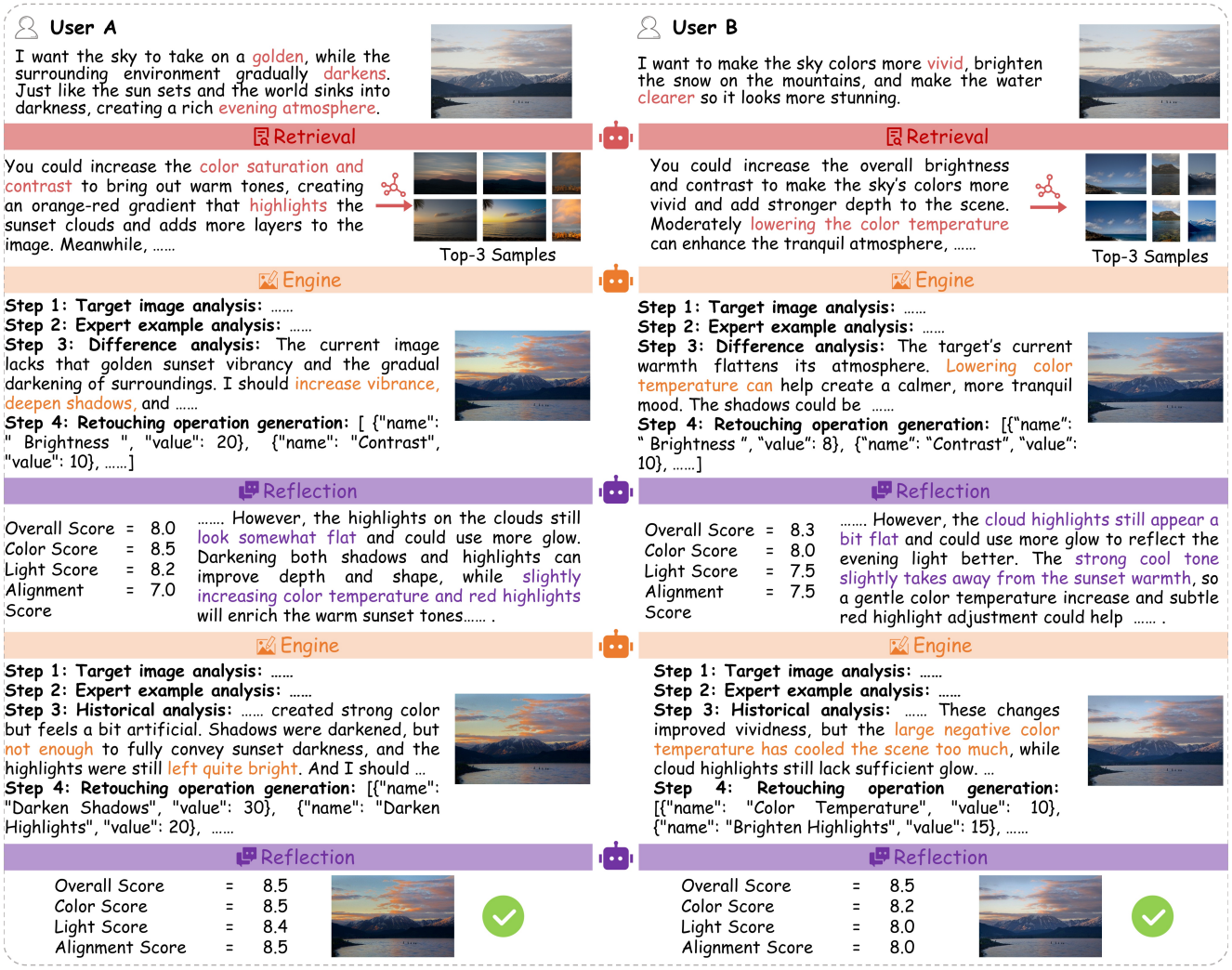

Figure 4: Two application cases of RetouchAgent. Given different requirements for the same image, our method provides distinct retrieval and retouching pipelines, enabling multi-round adjustments.

and RSFNet (Ouyang et al. 2023). Multi-style methods, including PIENet (Kim, Koh, and Kim 2020), StarEnhancer (Song, Qian, and Du 2021), TSFlow (Wang et al. 2024a), and DiffRetouch (Duan et al. 2025), are designed to handle diverse retouching preferences. The experimental results are shown in Table 1.

Overall, our method achieves either the best or competitive performance across all five experts, benefiting from the precise retrieval of exemplar images and the strong reasoning capability of MLLMs. Specifically, single-style methods suffer from significant performance drops when transferred to other styles, even though each subset is trained with a dedicated model. Our method also surpasses several recent white-box baselines, which can be attributed to the richer and more expressive operation library we designed. Finally, for multi-style methods, they are typically trained on data from all experts and adapt to specific styles by tuning different parameters. Among them, DiffRetouch performs comparably to our method in some cases, leveraging its powerful style transfer capabilities. However, our method consistently outperforms others in overall performance, particularly in terms of average metrics across all expert styles. More importantly, our case study in the following section confirms that RetouchAgent has a superior ability to interpret and respond to user intent expressed in natural language, a functionality that existing multi-style methods lack. Fig. 3 presents the qualitative comparison results on the MIT-Adobe FiveK. We can observe that our method produces retouching results that more closely align with expert edits, confirming its effectiveness in capturing expert-level retouching preferences.

We also compare our method with others on the PPR10k dataset, as shown in Table 2. While other methods are trained on separate subsets or all three experts, our method achieves superior performance using only a small number of expert examples.

3d-LUT	PIENet	DiffRetouch	Ours
8.4%	18.2%	28.6%	**44.8%**

Table 3: Comparison results of user study

λ_{text}	0	0.2	0.4	0.6	0.8
λ_{vis}	1	0.8	0.6	0.4	0.2
PSNR	23.50	25.78	**26.02**	25.82	25.48

Table 4: Ablation study on λ_{text} and λ_{vis}. The experiment is conducted on the subset of expert C.

User Study

In Fig. 4, we visualize the retouching processes of RetouchAgent applied to the same image with two different input requirements. For the same image, we guide the model to generate two distinct retouching styles: one evoking a sense of sunset glow, and the other conveying a serene atmosphere.

First, during the retrieval stage, RetouchAgent analyzes different requirements: Case 1 aims to match retouched samples with warm tones and a bright style, while Case 2 retrieves calm, low color temperature samples, resulting in two distinct sets of expert samples for prompt generation. Next, in the initial retouching stage, we highlight the comparative analysis between the input image and the expert examples. It is observed that in Case 1, RetouchAgent identifies substantial differences in vibrancy and applies bolder refinements. In contrast, in Case 2, the retrieved samples steer the agent to reduce the image's color temperature and focus more on balancing colors. Subsequently, in the reflection stage, the agent scores the outputs and analyzes previous iterations, enabling further retouching adjustments such as correcting excessively dark shadows and refining color curve details. Finally, after two rounds of adjustments, we obtain the two results for different requirements.

In addition, we conduct a user study to evaluate our method in real-world scenarios. Specifically, we compare the single-style method 3D-LUT, the multi-style method PIENet, and DiffRetouch. For 3D-LUT, we adopt the model trained on expert C. For PIENet, we follow (Kim, Koh, and Kim 2020) by using a personalized model adapted to user preferences. For DiffRetouch, we select the closest matching parameters according to the user's textual instructions. For our method, the retouching is guided exclusively by the user's textual instructions. We invite 40 users to evaluate 150 images sourced from MIT-Adobe FiveK and the internet, each associated with four different retouched results. The results are shown in Table 3. Our method achieved the highest user satisfaction without requiring any predefined preferences or manual parameter tuning.

Ablation Study

Ablation on image retrieval We conduct an ablation study to assess how different retrieval strategies affect model performance. As shown in Table 4, we report the results under different λ_{text} and λ_{vis}. Results show that image-only retrieval performs significantly worse than the multimodal retrieval strategy, highlighting the critical role of retouching intent in guiding effective image retouching. As the proportion of text increases, model performance improves initially. However, when textual content outweighs visual input, performance begins to decline. This is likely due to repeated patterns in generated retouching intents, which become more influential with excessive textual focus.

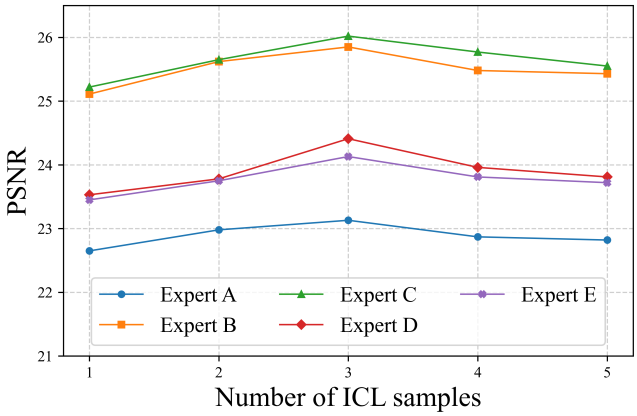

Figure 5: Ablation study on the number of ICL samples.

System Prompt	SR	System Prompt	SR
w/o	26.8%	w	**99.4%**

Table 5: Effectiveness of system prompts. 'w/o' means that we remove the constraints and examples for output standardization in the system prompt. The success rate is calculated across the entire MIT-Adobe FiveK dataset.

Next, we further conduct an ablation study on the number of ICL examples. Fig. 5 illustrates the PSNR performance of each of the five expert subsets under different numbers of ICl samples k. The results indicate that using three ICL examples yields the best performance. Too few samples fail to provide sufficient knowledge, whereas too many may introduce conflicting retouching instructions and increase the risk of model hallucinations.

Ablation on retouching engine The retouching engine consists of three key components: the system prompt, the operation library, and the CoT reasoning. The **system prompt** define the agent's role, provide retouching examples, and specify a standardized output format. Results in Table 5 show that this significantly improves the success rate of retouching. When system prompt are omitted, the successful rate (SR) of completing the entire workflow is only 26.8%.

The **operation library** offers the retouching operations that can be invoked with a single parameter, addressing diverse retouching demands. To validate its effectiveness, we conduct an ablation study by replacing it with two public alternatives: GMIC and OpenCV. GMIC offers extensive functionality but relies on multi-parameter, complex operations. OpenCV has simpler APIs but limited flexibility for detailed

Method	PSNR	SSIM
with GMIC	20.21	0.802
with OpenCV	21.65	0.823
w/o curve operations	23.12	0.903
Ours	**26.02**	**0.938**

Table 6: Ablation study on the operation library. The experiment is conducted on the subset of expert C.

Target	Expert	Difference	PSNR
✗	✓	✓	24.32
✓	✗	✓	24.16
✓	✓	✗	23.82
✓	✓	✓	**26.02**

Table 7: Ablation study on the CoT. The experiment is conducted on the subset of expert C.

control. We also test a reduced version 'w/o curve operations' by removing curve-level adjustments to simulate the operation sets in reinforcement learning-based methods. As shown in Table 6, both complex parameter configurations and simplified operation sets lead to performance degradation, highlighting the effectiveness of our operation library.

The **Chain-of-Thought** provides the agent with a structured mechanism for interpretation and operations generation. Table 7 further evaluates its impact on model performance. Results from Rows 1 and 2 show that explicit analysis of the original image and expert samples enables RetouchAgent to better understand retouching requirements and formulate appropriate strategies. Row 3 further demonstrates that fine-grained comparative analysis enhances the specificity of operation generation and improves retouching quality.

Ablation on reflection While MLLMs excel at multimodal reasoning, they struggle to produce high-quality retouching results in a single pass. To overcome this, we introduce a reflection agent that iteratively refines results using the history of previous steps. Table 8 presents the impact of incorporating the reflection module and different reflection strategies on model performance. Specifically, we compare the following settings: (1) without reflection, (2) continuous retouching based only on the result from the previous round (Round n-1), (3) retouching the original image at every round, and (4) our full method, which performs continuous retouching with the complete history of reflection.

The first two rows show that full reflection history improves performance, highlighting the reflection agent's role in enhancing the model's understanding of image retouching. Meanwhile, compared to restarting from the original image, iterative refinement proves more effective, as it better aligns with professional retouching workflows and reduces the cognitive load on the reflection agent.

Method	PSNR	SSIM
w/o reflection	21.35	0.816
only Round n-1	24.76	0.854
restart each round	24.84	0.862
Ours	**26.02**	**0.938**

Table 8: Ablation study on the reflection. The experiment is conducted on the subset of expert C.

Conclusion

In this work, we focus on the limitations of existing retouching methods regarding interactivity and explainability. To address this issue, we introduce RetouchAgent, the first framework to perform image retouching through collaboration among multiple MLLM agents. By building an intent-driven multimodal retrieval mechanism, we enable more efficient construction of ICL prompts. In addition to reduce hallucinations triggered by overly complex operations, we developed an operation library to support the retouching engine. Finally, we incorporated an iterative reflection module to better simulate the iterative nature of expert retouching practices. Quantitative experiments demonstrate that RetouchAgent achieves superior performance in understanding and executing expert retouching instructions. User studies further confirm its adaptability to diverse user needs, enabling personalized and high-quality retouching outcomes.

References

Agarwal, P.; Dave, H.; Bandlamudi, J.; Sindhgatta, R.; and Mukherjee, K. 2024. Multi-Stage Prompting for Next Best Agent Recommendations in Adaptive Workflows. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38: 22843–22849.

Bai, J.; Bai, S.; Yang, S.; Wang, S.; Tan, S.; Wang, P.; Lin, J.; Zhou, C.; and Zhou, J. 2023. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. arXiv:2308.12966.

Brooks, T.; Holynski, A.; and Efros, A. A. 2023. InstructPix2Pix: Learning to Follow Image Editing Instructions. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 18392–18402.

Brown, T. B.; Mann, B.; Ryder, N.; and Subbiah, e. a., Melanie. 2020. Language Models Are Few-Shot Learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 1877–1901.

Cao, M.; Wang, X.; Qi, Z.; Shan, Y.; Qie, X.; and Zheng, Y. 2023. MasaCtrl: Tuning-Free Mutual Self-Attention Control for Consistent Image Synthesis and Editing. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 22503–22513.

Dong, Q.; Li, L.; Dai, D.; Zheng, C.; Ma, J.; Li, R.; Xia, H.; Xu, J.; Wu, Z.; Chang, B.; Sun, X.; Li, L.; and Sui, Z. 2024. A Survey on In-Context Learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 1107–1128.

Duan, Z.-P.; Zhang, J.; Lin, Z.; Jin, X.; Wang, X.; Zou, D.; Guo, C.-L.; and Li, C. 2025. DiffRetouch: Using Diffusion to Retouch on the Shoulder of Experts. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39: 2825–2833.

Gao, Y.; Zhu, Q.; Fu, Y.; and Liang, D. 2024. Aesthetics-Driven Active Reinforcement Learning for Color Enhancement. In *Advanced Intelligent Computing Technology and Applications*, 289–300.

Gu, Z.; Zhu, B.; Zhu, G.; Chen, Y.; Tang, M.; and Wang, J. 2024. AnomalyGPT: Detecting Industrial Anomalies Using Large Vision-Language Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38: 1932–1940.

He, J.; Liu, Y.; Qiao, Y.; and Dong, C. 2020. Conditional Sequential Modulation for Efficient Global Image Retouching. *arXiv preprint arXiv:2009.10390*.

Hu, Y.; He, H.; Xu, C.; Wang, B.; and Lin, S. 2018. Exposure: A White-Box Photo Post-Processing Framework. *ACM Trans. Graph.*, 37: 26:1–26:17.

Ke, Z.; Sun, C.; Zhu, L.; Xu, K.; and Lau, R. W. H. 2022. Harmonizer: Learning to Perform White-Box Image and Video Harmonization. In *Computer Vision – ECCV 2022*, 690–706.

Kim, H.; and Lee, K. M. 2024. Learning Controllable ISP for Image Enhancement. *IEEE Transactions on Image Processing*, 33: 867–880.

Kim, H.-U.; Koh, Y. J.; and Kim, C.-S. 2020. PieNet: Personalized Image Enhancement Network. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX*, 374–390.

Kosugi, S.; and Yamasaki, T. 2020. Unpaired Image Enhancement Featuring Reinforcement-Learning-Controlled Image Editing Software. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34: 11296–11303.

Lee, S. H.; Jiang, J.; Xu, Y.; Li, Z.; Ke, J.; Li, Y.; He, J.; Hickson, S.; Datsenko, K.; Kim, S.; Yang, M.-H.; Essa, I.; and Yang, F. 2025. Cropper: Vision-Language Model for Image Cropping through In-Context Learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 30010–30019.

Li, B.; Li, X.; Lu, Y.; and Chen, Z. 2025. LossAgent: Towards Any Optimization Objectives for Image Processing with LLM Agents. arXiv:2412.04090.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023. BLIP-2: Bootstrapping Language-Image Pre-Training with Frozen Image Encoders and Large Language Models. arXiv:2301.12597.

Li, Y.; Xu, K.; Hancke, G. P.; and Lau, R. W. H. 2024a. Color Shift Estimation-and-Correction for Image Enhancement. In *2024 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 25389–25398.

Li, Z.; Zhang, F.; Cao, M.; Zhang, J.; Shao, Y.; Wang, Y.; and Sang, N. 2024b. Real-Time Exposure Correction via Collaborative Transformations and Adaptive Sampling. In *2024 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2984–2994.

Liang, J.; Zeng, H.; Cui, M.; Xie, X.; and Zhang, L. 2021. PPR10K: A Large-Scale Portrait Photo Retouching Dataset With Human-Region Mask and Group-Level Consistency. In *2021 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 653–661.

Liang, J.; Zeng, H.; and Zhang, L. 2021. High-Resolution Photorealistic Image Translation in Real-Time: A Laplacian Pyramid Translation Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9392–9400.

OpenAI; Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; and Aleman, e. a. 2024a. GPT-4 Technical Report. arXiv:2303.08774.

OpenAI; Hurst, A.; Lerer, A.; Goucher, A. P.; Perelman, A.; Ramesh, A.; and Clark, e. a. 2024b. GPT-4o System Card. arXiv:2410.21276.

Ouyang, W.; Dong, Y.; Kang, X.; Ren, P.; Xu, X.; and Xie, X. 2023. RSFNet: A White-Box Image Retouching Approach Using Region-Specific Color Filters. In *2023 Proceedings of the IEEE/CVF International Conference on Computer Vision*, 12160–12169.

Park, J.; Lee, J.-Y.; Yoo, D.; and Kweon, I. S. 2018. Distort-and-Recover: Color Enhancement Using Deep Reinforcement Learning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5928–5936.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, 8748–8763.

Song, Y.; Qian, H.; and Du, X. 2021. StarEnhancer: Learning Real-Time and Style-Aware Image Enhancement. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 4106–4115.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; Rodriguez, A.; Joulin, A.; Grave, E.; and Lample, G. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971.

Wang, H.; Zhang, J.; Liu, M.; Wu, X.; and Zuo, W. 2024a. Learning Diverse Tone Styles for Image Retouching. *IEEE Transactions on Image Processing*, 33: 310–321.

Wang, X.; Wang, W.; Cao, Y.; Shen, C.; and Huang, T. 2023. Images Speak in Images: A Generalist Painter for In-Context Visual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6830–6839.

Wang, Z.; Li, A.; Li, Z.; and Liu, X. 2024b. GenArtist: Multimodal LLM as an Agent for Unified Image Generation and Editing. In *Advances in Neural Information Processing Systems*, volume 37, 128374–128395.

Xue, W.; Ding, C.; Xu, R.; Wu, S.; Xu, Y.; and Wong, H.-S. 2025. RetouchGPT: LLM-Based Interactive High-Fidelity Face Retouching via Imperfection Prompting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39: 9059–9067.

Yan, Z.; Zhang, H.; Wang, B.; Paris, S.; and Yu, Y. 2016. Automatic Photo Adjustment Using Deep Neural Networks. *ACM Trans. Graph.*, 35: 11:1–11:15.

Yang, C.; Jin, M.; Jia, X.; Xu, Y.; and Chen, Y. 2022. AdaInt: Learning Adaptive Intervals for 3D Lookup Tables on Real-Time Image Enhancement. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 17501–17510.

Yang, S.; Huang, B.; Cao, M.; Ji, Y.; Guo, H.; Wong, N.; and Yang, Y. 2025. Taming Lookup Tables for Efficient Image Retouching. In *Computer Vision – ECCV 2024*, 144–159.

Zeng, H.; Cai, J.; Li, L.; Cao, Z.; and Zhang, L. 2022. Learning Image-Adaptive 3D Lookup Tables for High Performance Photo Enhancement in Real-Time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44: 2058–2073.

Zhang, F.; Zeng, H.; Zhang, T.; and Zhang, L. 2022. CLUT-Net: Learning Adaptively Compressed Representations of 3DLUTs for Lightweight Image Enhancement. In *Proceedings of the 30th ACM International Conference on Multimedia*, 6493–6501.

Zhang, S.; Yang, X.; Bai, X.; and Li, Y. 2024. Clip-Based Composition-Aware Image Cropping. In *2024 IEEE International Conference on Image Processing*, 2772–2778.

Zhou, Y.; Li, X.; Wang, Q.; and Shen, J. 2024a. Visual In-Context Learning for Large Vision-Language Models. arXiv:2402.11574.

Zhou, Y.; Li, X.; Wang, Q.; and Shen, J. 2024b. Visual In-Context Learning for Large Vision-Language Models. In *Findings of the Association for Computational Linguistics: ACL 2024*, 15890–15902. Bangkok, Thailand: Association for Computational Linguistics.