# RetouchGPT: LLM-based Interactive High-Fidelity Face Retouching via Imperfection Prompting

**Wen Xue[1]*, Chun Ding[1]*, Ruotao Xu[2], Si Wu[1,2]†, Yong Xu[1], Hau-San Wong[3]**

[1]School of Computer Science and Engineering, South China University of Technology
[2]Institute of Super Robotics (Huangpu)
[3]Department of Computer Science, City University of Hong Kong
{csxuewen, csdingchun}@mail.scut.edu.cn, rtxu@superobots.com
{cswusi, yxu}@scut.edu.cn, cshswong@cityu.edu.hk

## Abstract

Face retouching aims to remove facial imperfections from image and videos while at the same time preserving face attributes. The existing methods are designed to perform non-interactive end-to-end retouching, while the ability to interact with users is highly demanded in downstream applications. In this paper, we propose RetouchGPT, a novel framework that leverages Large Language Models (LLMs) to guide the interactive retouching process. Towards this end, we design an instruction-driven imperfection prediction module to accurately identify imperfections by integrating textual and visual features. To learn imperfection prompts, we further incorporate a LLM-based embedding module to fuse multi-modal conditioning information. The prompt-based feature modification is performed in each transformer block, such that the imperfection features are suppressed and replaced with the features of normal skin progressively. Extensive experiments have been performed to verify effectiveness of our design elements and demonstrate that RetouchGPT is a useful tool for interactive face retouching and achieves superior performance over state-of-the-arts.

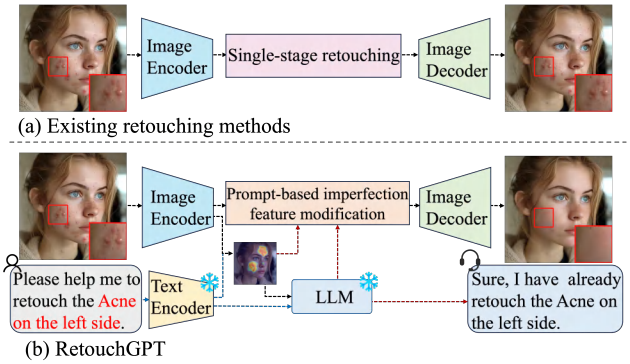


Figure 1: (a) Traditional face retouching frameworks that employ a single-stage face retouching framework lack the capability to comprehend user instructions. (b) Our RetouchGPT framework leverages Large Language Models (LLMs) to parse and interpret user instructions, enabling precise and interactive retouching.

## Introduction

The rapid advancement of social media has resulted in the generation and uploading of vast quantities of face images (Krizhevsky 2009; Liu et al. 2015) and videos (Zhu et al. 2021; Yu et al. 2023). Face retouching has emerged as a highly demanded visual task, to remove facial imperfections while preserving the face attributes. Normally people have different retouching preferences for various scenarios, such as virtual makeup try-on (Li et al. 2015), online meetings (Aseniero et al. 2020), and live streaming (Thang et al. 2014). However, the existing retouching methods perform non-interactive image-to-image translation, and thus cannot accommodate the specific instructions from users.

Traditional face retouching methods (Arakawa 2004; Velusamy et al. 2020) utilize nonlinear digital filters, which are only capable of removing minor blemishes on a small scale. To produce realistic clean faces, Generative Adversarial Networks (GANs) have been applied to the retouching task (Shafaei, Little, and Schmidt 2021; Xie et al. 2023). Furthermore, transformer architecture (Vaswani et al. 2017; Dosovitskiy et al. 2020) has also been utilized in this field (Wen et al. 2024; Xue et al. 2024), where the imperfections are removed by performing the cross-attention computation between the features of imperfections and normal skin. The existing methods are capable of single-stage retouching but are unable to refine the retouching results according to the user's feedback. On the other hand, the integration of Large Language Models (LLMs) with visual tasks has witnessed significant advancement. LLMs have the capability of understanding user instructions even in open-world scenarios. Models like LLaVA (Liu et al. 2024) and BLIP-2 (Li et al. 2023) attempt to bridge textual and visual modalities but typically fall short in fine-grained control. In addition, there are some attempts to integrate LLMs into downstream visual models (Wu et al. 2023; Gu et al. 2024), and leverage user instructions in fine-grained visual tasks. In this work, we facilitate face retouching by leveraging LLMs to understand user instructions and further control the generation process.

More specifically, we propose a novel retouching framework, RetouchGPT, which is the first attempt to leverage

*Joint first authors.
†Corresponding author.

LLMs for interactive face retouching. RetouchGPT is able to achieve user feedback-aligned and high-fidelity retouching results under the guidance of LLM. To precisely locate facial imperfections, we incorporate an Instruction-driven Imperfection Prediction (IIP) module that learns to match the instruction and visual features and returns imperfection masks. Furthermore, we designed an LLM-Based Embedding (LBE) module to learn imperfection prompts by fusing the obtained textual and visual conditioning information. To guide the content generation in the imperfection regions, the resulting prompts are injected into a latent transformer, and imperfection feature modification is performed via cross-attention in each block. We have verified the effectiveness of the proposed IIP and LBE in imperfection prediction and multi-modal information fusion and performed a comprehensive comparison with state-of-the-art methods in a variety of face retouching tasks. The main contributions are as follows: **(a)** Different from the existing methods that perform single-stage face retouching, the proposed RetouchGPT is capable of interactive retouching by working together with LLM. **(b)** By integrating user's instruction and visual features, facial imperfection prediction performance can be improved significantly. **(c)**We fuse multi-modal conditioning information via LLM to obtain imperfection prompts, which controls imperfection feature modification in the interactive retouching process.

## Related Work

### Face Retouching

Face retouching aims to enhance facial images by removing imperfections and improving overall aesthetic appeal. Traditional methods (Arakawa 2004; Velusamy et al. 2020; Batool and Chellappa 2014) leveraged simple image processing techniques such as nonlinear digital filters, smoothing operators, texture orientation fields, and Markov random field modeling, which are limited to addressing specific types of imperfections, such as small scale wrinkles or spots. With the advent of deep learning (Krizhevsky, Sutskever, and Hinton 2012; He et al. 2016), AutoRetouch (Shafaei, Little, and Schmidt 2021) utilized generative adversarial networks (GANs) (Goodfellow et al. 2020; Karras, Laine, and Aila 2019) for image-to-image translation, which can learn detailed patterns and differentiate between blemishes and natural skin features, thereby targeting various imperfection types. Adaptive Blend Pyramid Network (ABPN)(Lei et al. 2022) was proposed to realize fast local retouching on high-resolution photos by utilizing an adaptive blend pyramid structure to handle various imperfections types. Furthermore, to further achieve the performance of large-scale imperfection retouching, BPFRe (Xie et al. 2023) introduced an attention module that learns to infer a blemish-aware map and further determines the corresponding weights, thus performing progressive blemish removal.

In addition, researchers found that the attention mechanism in the transformer can focus on the imperfections, and proposed a number of transformer-based frameworks for face retouching. For example, RetouchFormer (Wen et al. 2024) formulates face retouching as a 'soft inpainting' task, using a transformer with a selective cross-attention mechanism to synthesize clean face images with high realism and fidelity. To handle the face video retouching task, VRetochEr (Xue et al. 2024) leverages temporal context information to facilitate face retouching in dynamic sequences. Specifically, it performs imperfection flow estimation to obtain displacement information between consecutive frames, refining imperfection localization and ensuring stable retouching performance across frames. Nevertheless, the current retouching techniques are unable to produce optimal results when confronted with unknown imperfections. In particular, they encounter difficulties when confronted with imperfections that are not included in the training data and are unable to incorporate instructions, which leads to unsatisfactory retouching outcomes.

### Large Language Model

Large Language Models (LLMs) have demonstrated enhanced capabilities in language comprehension and reasoning, due to their efficient transformer-based architecture. Accordingly, LLMs are well suited for handling challenging tasks across a wide range of application domains. The success of the transformer-based architecture has led to the advent of pre-trained models with larger parameter scales, including BERT (Devlin et al. 2019), T5 (Raffel et al. 2020; Chung et al. 2024), and the GPT series (Radford et al. 2018, 2019; Brown et al. 2020). The advent of general LLMs, such as ChatGPT (Ouyang et al. 2022; Achiam et al. 2023) and Llama (Touvron et al. 2023a,b; Dubey et al. 2024), represents a major advancement in the field of language understanding and generation.

To improve the performance of real-world visual tasks, researchers have proposed various visual LLMs. For instance, LLaVA (Liu et al. 2024) employed a visual encoder to extract image features and connect them to the word embedding space, which enables the LLM to process both textual and visual information for multi-modal tasks. Moreover, BLIP-2 (Li et al. 2023) transforms images into text-based queries, augmenting LLM's understanding of visual information. Additionally, Visual ChatGPT (Wu et al. 2023) employed a prompt manager that translates diverse visual inputs into linguistic formats. With regard to the industrial anomaly detection task, AnomalyGPT (Gu et al. 2024), which focuses on the industrial anomaly detection task, employed vision-textual feature matching decoder to enhance the semantic recognition capabilities and fine-tuned with prompt embedding, improving anomaly detection efficacy.

Different from the existing retouching methods, RetouchGPT **(1)** integrates textual and visual information to accurately predict imperfections, **(2)** leverages LLMs to fuse multi-modal information and translate it into imperfection prompts, and **(3)** performs prompt-based imperfection feature modification to suppress imperfections.

## Proposed Method

### Problem Settings

In this section, we introduce the methodology of RetouchGPT for interactive face retouching, which aims to pro-

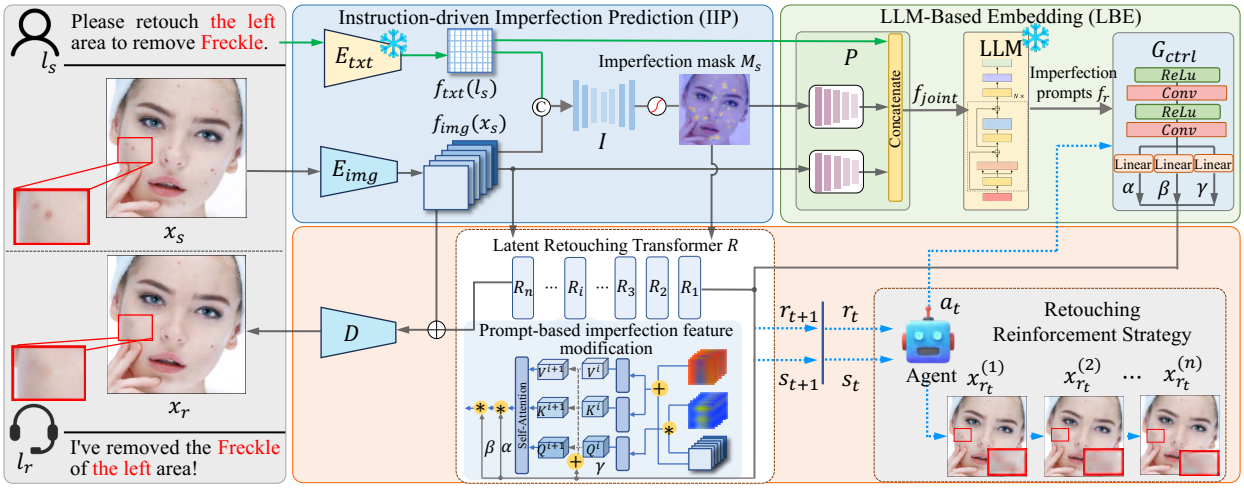


Figure 2: The workflow of the proposed interactive face retouching framework: RetouchGPT. In IIP, image encoder $E_{img}$ and text encoder $E_{txt}$ extract features $f_{img}(x_s)$ and $f_{txt}(l_s)$ from the source image $x_s$ and natural language text $l_s$, and the imperfection prediction network $I$ then generates imperfection mask $M_s$. We utilized LLM in LBE, a prompt learner $P$ combines these features and the imperfection mask $M_s$ to create a multi-modal joint conditioning embedding $f_{joint}$. The LLM fuses the multi-modal conditioning information to generate the imperfection prompts $f_r$, which are then employed to control the generator $G_{ctrl}$ in producing conditioning embedding $\{\alpha, \beta, \gamma\}$. Using $\{\alpha, \beta, \gamma\}$ and $M_s$, the latent retouching transformer $R$ performs prompt-based imperfection feature modification. The decoder $D$ finally generates the retouched face image $x_r$. In addition, a retouching reinforcement strategy is designed and adopted to further improve the iterative retouching process.

gressively improve the quality of face retouching according to user feedback. The problem can be formalized as an optimization task, and the goal is to generate a retouched image $x_r$ that closely matches the user instruction $l_u$ under certain standards. This problem can be formulated as follows:

$$x_r \leftarrow \arg\min_{x_r} \mathbb{E}[\|\phi\left(f_{img}(x_r), f_{txt}(l_u)\right)\|], \tag{1}$$

where $\|\cdot\|$ denotes the norm used to measure the difference between the retouched image $x_r$ and the user instruction $l_u$. The function $\phi(\cdot,\cdot)$ maps the image feature $f_{img}(x_r)$ and the user instruction feature $f_{txt}(l_u)$ into the same feature space. To achieve this, we incorporate an Instruction-driven Imperfection Prediction (IIP) module and an LLM-Based Embedding (LBE) module to perform prompt-based imperfection feature modification.

## Instruction-driven Imperfection Prediction

Given a source image $x_s$ and its corresponding text description $l_s$, the image and text feature encoders $\{E_{img}, E_{txt}\}$ are utilized to extract features $\{f_{img}(x_s), f_{txt}(l_s)\}$, which are fed into the imperfection prediction network $I$ to predict imperfection mask $M_s$. The network $I$ identify facial imperfections from source image and its features $\{f_{img}(x_s), f_{txt}(l_s)\}$, and can be represented as:

$$M_s = \sigma\left(I\left[x_s, \left(f_{img}(x_s) \oplus f_{txt}(l_s)\right)\right]\right), \tag{2}$$

where $\sigma(\cdot)$ denotes an learnable activation function, and $\oplus$ represents the concatenate process. The imperfection mask $M_s$, which indicates the locations of imperfections, guides latent retouching transformer to suppress and modify imperfection features based on normal facial features. By leveraging the IIP module, RetouchGPT can generate imperfection masks that align with user feedback, thereby achieving optimal imperfection prediction and further improving the retouching performance.

## LLM-Based Embedding

The LLM-Based Embedding (LBE) module is designed to generate imperfection prompts $f_r$, conditioned on the imperfection mask $M_s$ and features $f_{img}(x_s), f_{txt}(l_s)$. Specifically, we design a prompt learner $P$ to fuse and align multi-modal imperfection prompting. $P$ receives the source image $x_s$, text description $l_s$, and imperfection mask $M_s$ as input conditional multi-model information, subsequently producing a joint embedding $f_{joint}$. This joint embedding $f_{joint}$ encapsulates rich multi-modal imperfection information, enabling LLM to generate the corresponding imperfection prompts $f_r$ based on its understanding of the imperfection information, which can be formulated as follows:

$$f_r = \text{LLM}\left(P\left(f_{img}(x_s), f_{txt}(l_s), M_s\right)\right). \tag{3}$$

Next, we design a conditioning embedding generator $G_{ctrl}$ to compute the conditioning embedding $\{\alpha, \beta, \gamma\}$ based on the imperfection prompts $f_r$:

$$\{\alpha, \beta, \gamma\} = G_{ctrl}\left(\sigma(f_r)\right), \tag{4}$$

where $G_{ctrl}$ computes the conditioning embeddings $\{\alpha, \beta, \gamma\}$ to be used in prompt-based imperfection feature modification. These conditioning embeddings incorporate rich multi-modal conditional controlling information. Our LBE module aims to utilize LLM to fuse the multi-modal information and generate imperfection prompts, therefore controlling the calculation of cross-attention to achieve interactive retouching according to user feedback.

## Prompt-based Imperfection Feature Modification

To realize interactive retouching, we propose prompt-based imperfection feature modification associated with the latent retouching transformer $R$. The input features $f_{img}(x_s)$, imperfection mask $M_s$, and conditioning embedding $\{\alpha, \beta, \gamma\}$ are fed into the latent retouching transformer $R$, which is defined as follows:

$$\begin{aligned} Q^i =& W_q(R^{i-1} \otimes M_s) + \gamma, \\ \{K^i, V^i\} =& W_{kv}\left(R^{i-1} \otimes (1 - M_s)\right) + \gamma, \end{aligned} \tag{5}$$

where $R^i$ denotes the retouched feature of the $i$-th transformer block. RetouchGPT uses the imperfection map $M_s$ and its complement $(1 - M_s)$ as a soft mask to enhance regions to be edited, with $M_s$ and $(1-M_s)$ representing masks for imperfection and normal regions, respectively. The reorganized attention vectors $\{Q^i, K^i, V^i\}$ then apply a cross-attention, which aims to suppress the expression of imperfections using features of normal facial regions:

$$\begin{aligned} R^i &= \alpha \cdot \text{softmax}\left(Q^i \cdot K^i/\lambda\right) V^i + \beta, \\ x_r &= D\left(f_{img}(x_s) \otimes (1 - M_s) + R^n \otimes M_s\right), \end{aligned} \tag{6}$$

where $R^n$ is the output of the final block, $\lambda$ denotes the scaling factor to ensure numerical stability. RetouchGPT generates the retouched image $x_r$ using an image decoder $D$. This process is controlled by the conditioning embedding $\{\alpha, \beta, \gamma\}$ and the imperfection mask $M_s$.

## Model Optimization

The training process includes the joint optimization of three components: imperfection prediction, LLM-based embedding, and retouched image generation. Let $x_s$ and $x_t$ represent the source raw image and the retouched target image, $l_s$ and $l_{gt}$ denote the user instruction and the target response, respectively. We consider that the difference between $x_s$ and $x_t$ indicates the imperfections needed to be retouched, which is denoted as $M_{gt}$. RetouchGPT takes $x_s$ and $l_s$ as input and generates the retouched image $x_r$, response $l_r$ and imperfection mask $M_s$. The precision of imperfection prediction in the IIP module is evaluated by measuring the discrepancy between the prediction $M_s$ and the ground truth imperfections $M_{gt}$. The corresponding loss function $\mathcal{L}_{mask}$ is formulated as follows:

$$\mathcal{L}_{mask} = \mathbb{E}_{x_s}\left[\|M_s - M_{gt}\|\right]. \tag{7}$$

To further optimize the imperfection prompts inside our LBE module, we utilize the cross-entropy loss to measure the disparity between the generated text sequence $l_r$ and the target text sequence $l_{gt}$:

$$\mathcal{L}_{embed} = \mathbb{E}_{l_s}\left[-\sum_{i=1}^{n} l_{gt}^i \log({l_r}^i)\right], \tag{8}$$

where $n$ is the number of tokens, $l_{gt}^i$ is the true label for token $i$, and $l_r^i$ is the predicted probability for token $i$. To further constrain the generated image $x_r$, we employ L1 regression to measure the degree of consistency between the target $x_t$ and the model's output $x_r$ as follows:

$$\mathcal{L}_{retouch} = \mathbb{E}_{x_s}\left[\kappa\|x_r - x_t\|_1 + \|\mathcal{H}(x_r) - \mathcal{H}(x_t)\|_2^2\right], \tag{9}$$

where $\kappa$ denotes a weighting factor, and $\mathcal{H}(\cdot)$ represents the features extracted from a pre-trained VGG-19 (Simonyan and Zisserman 2014). We also designed a Retouching Reinforcement Strategy for prompt-driven retouching based on the user's instructions.

**Retouching Reinforcement Strategy.** RetouchGPT utilizes training texts $l_s$ containing imperfection position $m_p$ and imperfection category $m_c$ for interactive retouching. The state $s_t$ at time $t$ is defined by the feature representation $f_{img}(x_s)$ of the current retouched image, the imperfection $M_s$, and the joint embedding $f_{joint}$. The action $a_t$ involves the retouching operation by transformer $R$, which includes calculating conditioning embeddings $\{\alpha, \beta, \gamma\}$ and performing progressive-retouching operations on the image. This process generates multiple images, represented as $\left\{x_{r_t}^{(i)} = F\left(x_s, \alpha^{(i)}, \beta^{(i)}, \gamma^{(i)}\right)\right\}_{i=1}^{n}$, where $n$ is the number of operations, and $F(\cdot,\cdot,\cdot,\cdot)$ performs progressive-retouching operations. The reward $r_t$ is based on the consistency between the retouched image $x_{r_t}$ and the training text $l_s$, measured by:

$$r_t = \psi(x_{r_t}, l_s), \tag{10}$$

where $\psi(\cdot,\cdot)$ calculates the alignment between the image and text. For each training sample $\{x_s, l_s, m_p\}$, $x_{r_0}$ denotes the initial generated retouched image. In each iteration, the current state $s_t$ includes $f_E(x_s)$, $M_s$, and $f_{joint}$. A new retouched image $x_{r_t}$ is generated based on action $a_t$, and the reward $r_t$ is calculated by evaluating the alignment with $l_s$. The value function $V(s_t)$ represents the expectation of cumulative reward starting from state $s_t$, considering all possible subsequent actions $\{a_{t+1}, a_{t+2}, \dots\}$ (Mnih et al. 2013). It is updated using temporal difference learning (Rasoul, Adewole, and Akakpo 2021):

$$V(s_t) \leftarrow V(s_t) + \zeta\left[r_{t+1} + \eta V(s_{t+1}) - V(s_t)\right], \tag{11}$$

where $V(s_t)$ and $V(s_{t+1})$ are the values of the current and next states, $r_{t+1}$ is the next reward, and $\zeta$, $\eta$ are the learning rate and discount factor. The reinforcement learning loss function $\mathcal{L}_{RL}$ is defined as:

$$\mathcal{L}_{RL} = \mathbb{E}_{(a_t, s_t, r_t)}\left[\left(r_{t+1} + \eta V(s_{t+1}; \theta) - V(s_t; \theta)\right)^2\right], \tag{12}$$

where $\theta$ represents the model parameters. Finally, we integrate the above three aspects, and express the optimization formulation of the proposed approach as follows:

$$\begin{aligned} &\min_{E_{img}, I} \ \mathcal{L}_{mask}, \\ &\min_{P, G_{ctrl}} \ \mathcal{L}_{embed}, \\ &\min_{E_{img}, R, D} \ \mathcal{L}_{retouch} + \mathcal{L}_{RL}. \end{aligned} \tag{13}$$

It should be noted that the constituent networks are optimized with different loss terms $\mathcal{L}_{mask}, \mathcal{L}_{embed}$ and $\mathcal{L}_{retouch}$. In addition, the reinforcement strategy fine-tunes RetouchGPT with $\mathcal{L}_{RL}$ at the end of each epoch, guiding the overall retouching quality. The goal of this integrated training approach is to achieve high-fidelity, interactive face retouching that is aligned with user feedback. Empirically, we find that our RetouchGPT can converge to a good solution without heavy tuning.

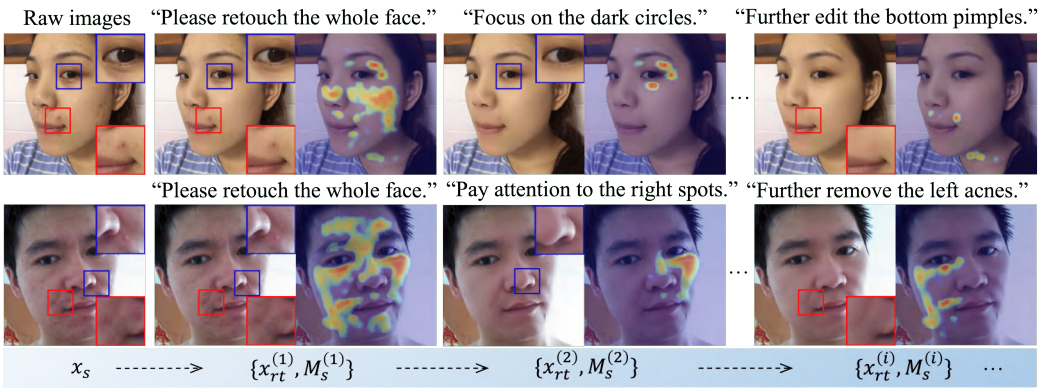


Figure 3: The visual results of interactive retouching using different instructions. $x_{rt}^{(i)}$ and $M_s^{(i)}$ denote the $i$-th retouching interactive result and corresponding soft mask, respectively. RetouchGPT performs multiple retouching operations according to the proposed reinforcement strategy.

# Experiments

## Experiment Settings

**Datasets.** We utilized Flickr-Face-HQ-Retouching dataset (FFHQR) (Shafaei, Little, and Schmidt 2021) (contains 56k/7k/7k train/evaluate/test images) for comparison. Additionally, we collected 1.3k challenging test samples from FFHQR to compose the FFHQR-hard subset. With the assistance of ChatGPT-4o (OpenAI 2024), we labeled imperfection categories, positions and corresponding user instructions on FFHQR and 1k in-the-wild images.

**Training Details.** In the training process, the parameters of RetouchGPT are updated by the Adam optimizer (Kingma and Ba 2015) with the learning rate of $2\times10^{-4}$. The hyper-parameter $\kappa, \zeta, \eta$ are set to 10, 0.1, and 0.9, respectively. We use the pre-trained T5 model (Raffel et al. 2020) as text encoder and Llama (Touvron et al. 2023b) as LLM. They will be frozen during training. There are a total of 400k training iterations, and the batch size is set to 1. We implement RetouchGPT by using PyTorch and train it on a single GPU with 80G graphics memory.

**Evaluation Protocols.** All competing methods are implemented using open-source codes. To assess the consistency between the synthesized image and the target image, we use widely accepted metrics: PSNR, SSIM, LPIPS (Zhang et al. 2018), VFID and Soft-IoU. We also adopt ROUGE-1 (Lin 2004) to evaluate text generation performance.

## Imperfection Prediction Analysis

To verify the effectiveness of IIP, we compare RetouchGPT with a number of other models: BPFRe, RetouchFormer. Additionally, we disabled $E_{txt}$ in the IIP module, denoted as *w/o* $E_{txt}$, to assess the significance of text embedding. As shown in Table 1, under the prompt 'Predict the imperfections precisely', RetouchGPT demonstrates superior performance in predicting imperfections. We perform a visual comparison of imperfection detection in Figure 4, RetouchGPT can locate various imperfections precisely, including pimples, dark circles, fine lines, and so on. The comparison between RetouchGPT *w/o* $E_{txt}$ and ours demonstrates the importance of user input for effective imperfection prediction. In addition, we visualize the imperfection map $M_s$ from Eq. (2) under different prompts, as illustrated in Figure 3, which effectively highlights RetouchGPT's capability to accurately focus on the specified regions based on the given prompts.

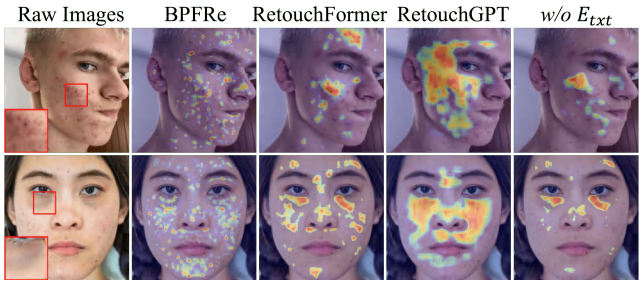


Figure 4: Visual comparison in imperfection prediction. User instruction: 'Predict the imperfections of whole face'.

| Methods | SSIM ↑ | Soft-IoU ↑ | ROUGE-1(%) ↑ |
|---|---|---|---|
| BPFRe | 0.6012 | 0.2903 | - |
| RetouchFormer | 0.7192 | 0.4155 | - |
| AnomalyGPT | 0.7046 | 0.4398 | - |
| *w/o* $E_{txt}$ | 0.7181 | 0.4119 | 37.02 |
| RetouchGPT | **0.8418** | **0.4723** | **89.72** |

Table 1: Quantitative comparison in imperfection prediction. RetouchGPT utilize textual encoder to achieve more precise imperfection location performance.

## Discussion on Interactive Retouching

As shown in Figure 3, we used different instructions and realized a progressive-retouching operation. RetouchGPT demonstrates an understanding of these instructions, accurately predicting the corresponding imperfections and re-

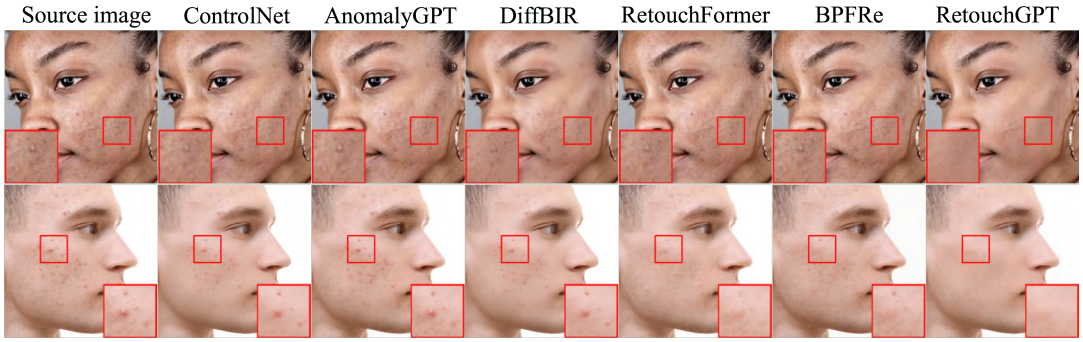


Figure 5: Representative retouching results of RetouchGPT and competing methods on in-the-wild face images.

| Methods | User Study | ChatGPT4-o |
|---|---|---|
| AutoRetouch | 4.09 ± 2.05 | 5.56 ± 1.74 |
| BFPRe | 5.05 ± 1.71 | 6.17 ± 0.67 |
| RetouchFormer | 5.93 ± 0.72 | 6.76 ± 0.51 |
| VRetouchEr | 5.84 ± 0.68 | 6.25 ± 0.37 |
| RetouchGPT *w/o IIP & LBE* | 4.28 ± 1.84 | 5.93 ± 1.63 |
| RetouchGPT *w/o IIP* | 5.67 ± 1.42 | 6.36 ± 0.92 |
| RetouchGPT *w/o LBE* | 6.05 ± 0.70 | 6.87 ± 0.43 |
| RetouchGPT | **8.73 ± 0.45** | **8.11 ± 0.35** |

Table 2: Comparison of RetouchGPT and competing methods in terms of user/ChatGPT4-o's rating score. A higher score indicates superior model performance, with a maximum score of 10.

touching the image based on the given instructions. Additionally, we compare the interactive accuracy in Table 2. A user study involving over 100 participants was conducted to assess the retouching performance and accuracy of the competing methods with a full score of 10. We use the ChatGPT4-o (OpenAI 2024) as the evaluation model to assess retouching performance and accuracy with a full score of 10. As shown in Table 2, RetouchGPT can accurately detect and retouch images based on user instructions, resulting in superior retouching outcomes compared to other methods.

## Discussion on Designed Elements

We discuss the effectiveness of IIP and LBE. To this end, we first implement a model only using the encoder $E_{img}$, transformer $R$ with cross-attention and $D$, denoted as RetouchGPT *w/o IIP & LBE*. As shown in Table 2, the lack of instruction-driven imperfection prediction and the imperfection prompts of LLM lead to a significant reduction in retouching performance.

**(1) Does the imperfection mask from IIP make sense?** We leverage IIP to learn imperfection mask $M_s$ by integrating visual and textual features$\{f_{img}(x_s), f_{txt}(l_s)\}$. To verify the effectiveness of $M_s$, we obtain a variant 'RetouchGPT *w/o IIP*' by removing the IIP and $M_s$ and using self-attention(Vaswani et al. 2017) in Eq. (5) & (6). The results shown in Table 2 suggest that $M_s$ brings about 26.9% performance gains from user feedback. As to the visual results in Figure 6, we can observe that $M_s$ is crucial for identifying and retouching the specified imperfections.

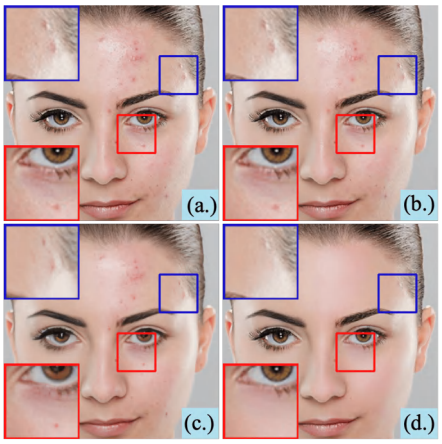


Figure 6: Representative visual results of different design components. From (a.) to (b.) are RetouchGPT *w/o IIP & LBE*, RetouchGPT *w/o IIP*, RetouchGPT *w/o LBE*, and RetouchGPT, respectively. RetouchGPT is able to suppress detailed imperfections (such as a red spot under the right eye).

**(2) Is the LBE module important?** We integrate LLM in LBE to fuse the multi-modal imperfection information enabling our RetouchGPT to learn imperfection prompts $f_r$, thereby effectively enhancing the efficacy and interactivity of retouching. To verify the effectiveness of LBE, we build a variant RetouchGPT *w/o LBE*, by removing $P$, $G_{ctrl}$ and the conditioning embedding $\{\alpha, \beta, \gamma\}$ in Eq. (4) & (5) & (6). As shown in Figure 6, RetouchGPT *w/o LBE* can realize an overall satisfactory retouching quality but fails in retouching the details. Table 2 shows that *w/o LBE* leads to a performance drop of 1.24 in terms of ChatGPT4-o score.

**(3) Are user instructions meaningful in interactive retouching?** To evaluate the controllability of RetouchGPT, we visualized the imperfection prompts $f_r$ of Eq. (3) & (4) from LLM in Figure 7. According to different user instructions, $f_r$ forms multiple clusters in the latent space, indicating that $f_r$ is sensitive to the user's instructions, and is capable of identifying different categories and positions of

| Methods | FFHQR | | FFHQR-hard | |
|---|---|---|---|---|
| | PSNR ↑ | LPIPS ↓ | PSNR ↑ | LPIPS ↓ |
| AutoRetouch | 44.18 | 0.0365 | 37.18 | 0.0332 |
| ControlNet | 40.17 | 0.0528 | 35.21 | 0.0943 |
| BPFRe | 45.12 | 0.0091 | 38.83 | 0.0213 |
| DiffBIR | 41.57 | 0.0893 | 36.80 | 0.1032 |
| RetouchFormer | 45.66 | 0.0071 | 38.94 | 0.0157 |
| AnomalyGPT | 41.35 | 0.0722 | 35.61 | 0.1127 |
| VRetouchEr | 45.39 | 0.0102 | 38.79 | 0.0198 |
| **RetouchGPT** | **46.21** | **0.0023** | **39.68** | **0.0115** |

Table 3: Quantitative retouching results of RetouchGPT and competing methods on FFHQR and FFHQR-hard.

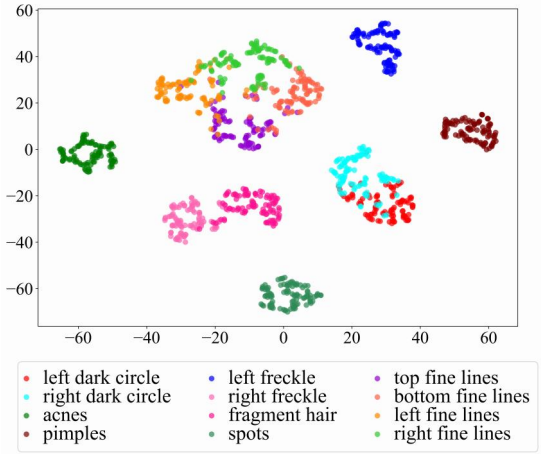


Figure 7: Visualizing the distributions of $f_r$ using different instructions.

facial imperfections. Therefore, as shown in Figure 3, different user instructions can lead to multiple retouching operations. This indicated that user instructions can provide detailed guidance to perform interactive retouching.

## Comparisons to State-of-the-arts

We compare our RetouchGPT with existing retouching methods, including the GAN-based methods AutoRetouch (Shafaei, Little, and Schmidt 2021) and BPFRe (Xie et al. 2023), diffusion-based methods ControlNet (Zhang, Rao, and Agrawala 2023) and DiffBIR (Lin et al. 2023), transformer-based RetouchFormer (Wen et al. 2024), and the LLM-based AnomalyGPT (Gu et al. 2024). We leverage 'Please retouch the whole image with high-fidelity' as user instruction. As shown in Table 3, BPFre, RetouchFormer and VRetouchEr achieve similar performance, diffusion-based methods DiffBIR, and ControlNet perform less satisfactorily. RetouchFormer outperforms all the other comparing motheds while our RetouchGPT achieves a higher PSNR score by 0.55 dB. RetouchGPT can also achieve better results than the competing methods in terms of LPIPS. Visual comparisons in Figure 5 indicate that competing methods are unstable with unseen imperfections, while RetouchGPT can achieve stable high-fidelity retouching performance.

| Methods | PSNR ↑ | SSIM ↑ | LPIPS ↓ | VFID ↓ |
|---|---|---|---|---|
| BFPRe | 38.14 | 0.9711 | 0.0275 | 17.81 |
| RetouchFormer | 38.69 | 0.9774 | 0.0219 | 12.36 |
| VRetouchEr | 39.55 | 0.9813 | 0.0191 | 10.01 |
| RetouchGPT | **40.12** | **0.9878** | **0.0169** | **7.01** |

Table 4: Quantitative comparison in face video retouching.

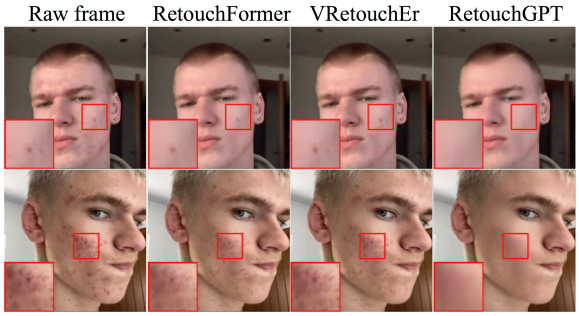


Figure 8: Representative retouching results on video frames.

## Extension to Face Video Retouching

In this section, we compare RetouchGPT, RetouchFormer, and VRetouchEr for face video retouching. We use 'Please retouch the frames stably with high quality' as user instruction for RetouchGPT, and follow the FFHQR-seq test set protocol (Xue et al. 2024). As shown in Figure 8, RetouchGPT achieves consistently stable performance without specific video retouching training. The quantitative results in Table 4 show that RetouchGPT surpasses VRetouchEr, demonstrating its robustness and generalization in face video retouching.

## Conclusion

In this work, we introduce RetouchGPT, the first attempt to leverage LLM for interactive face retouching. By integrating user instructions with visual features, we have significantly improved the performance of facial imperfection prediction. Furthermore, we integrate LLMs in RetouchGPT to fuse multi-modal conditioning information and consequently perform imperfection prompt-based feature modification to realize interactive and high-fidelity retouching. Extensive comparisons demonstrate the interactivity and superior performance of RetouchGPT.

## Acknowledgements

This work was supported in part by the Key Realm Research and Development Program of Guangzhou (Project No. 2024B01W0007), in part by the National Natural Science Foundation of China (Project No. 62072189), in part by the GuangDong Basic and Applied Basic Research Foundation (Project No. 2024A1515011437), and in part by TCL Science and Technology Innovation Fund (Project No. 20231752).

## References


Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Arakawa, K. 2004. Nonlinear digital filters for beautifying facial images in multimedia systems. In *2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No. 04CH37512)*, volume 5, V–V. IEEE.

Aseniero, B. A.; Constantinides, M.; Joglekar, S.; Zhou, K.; and Quercia, D. 2020. MeetCues: Supporting online meetings experience. In *2020 IEEE Visualization Conference (VIS)*, 236–240. IEEE.

Batool, N.; and Chellappa, R. 2014. Detection and inpainting of facial wrinkles using texture orientation fields and Markov random field modeling. *IEEE transactions on image processing*, 23(9): 3773–3788.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, 33: 1877–1901.

Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70): 1–53.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144.

Gu, Z.; Zhu, B.; Zhu, G.; Chen, Y.; Tang, M.; and Wang, J. 2024. Anomalygpt: Detecting industrial anomalies using large vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 1932–1940.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4401–4410.

Kingma, D. P.; and Ba, J. L. 2015. Adam: a method for stochastic optimization. In *Proc. International Conference on Learning Representation*.

Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images. *Master's thesis, University of Tront*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Lei, B.; Guo, X.; Yang, H.; Cui, M.; Xie, X.; and Huang, D. 2022. Abpn: Adaptive blend pyramid network for real-time local retouching of ultra high-resolution photo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2108–2117.

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. In *International conference on machine learning*, 19730–19742. PMLR.

Li, J.; Xiong, C.; Liu, L.; Shu, X.; and Yan, S. 2015. Deep face beautification. In *Proceedings of the 23rd ACM international conference on Multimedia*, 793–794.

Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, 74–81. Barcelona, Spain: Association for Computational Linguistics.

Lin, X.; He, J.; Chen, Z.; Lyu, Z.; Fei, B.; Dai, B.; Ouyang, W.; Qiao, Y.; and Dong, C. 2023. Diffbir: Towards blind image restoration with generative diffusion prior. *arXiv preprint arXiv:2308.15070*.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2024. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, 3730–3738.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

OpenAI. 2024. Hello GPT-4o. https://openai.com/index/hello-gpt-4o/. Accessed: 2024-5-13.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pre-training. *OpenAI blog*. Publisher: OpenAI.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140): 1–67.

Rasoul, S.; Adewole, S.; and Akakpo, A. 2021. Feature selection using reinforcement learning. *arXiv preprint arXiv:2101.09460*.

Shafaei, A.; Little, J. J.; and Schmidt, M. 2021. Autoretouch: Automatic professional face retouching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 990–998.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Thang, T. C.; Le, H. T.; Pham, A. T.; and Ro, Y. M. 2014. An evaluation of bitrate adaptation methods for HTTP live streaming. *IEEE Journal on Selected Areas in Communications*, 32(4): 693–705.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, 6000–6010. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781510860964.

Velusamy, S.; Parihar, R.; Kini, R.; and Rege, A. 2020. FabSoften: Face beautification via dynamic skin smoothing, guided feathering, and texture restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 530–531.

Wen, X.; Xie, L.; Jiang, L.; Chen, T.; Wu, S.; Liu, C.; and Wong, H.-S. 2024. RetouchFormer: Semi-supervised High-Quality Face Retouching Transformer with Prior-Based Selective Self-Attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 5903–5911.

Wu, C.; Yin, S.; Qi, W.; Wang, X.; Tang, Z.; and Duan, N. 2023. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*.

Xie, L.; Xue, W.; Xu, Z.; Wu, S.; Yu, Z.; and Wong, H. S. 2023. Blemish-aware and Progressive Face Retouching with Limited Paired Data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5599–5608.

Xue, W.; Jiang, L.; Xie, L.; Wu, S.; Xu, Y.; and Wong, H. S. 2024. VRetouchEr: Learning Cross-frame Feature Interdependence with Imperfection Flow for Face Retouching in Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9141–9150.

Yu, J.; Zhu, H.; Jiang, L.; Loy, C. C.; Cai, W.; and Wu, W. 2023. Celebv-text: A large-scale facial text-video dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14805–14814.

Zhang, L.; Rao, A.; and Agrawala, M. 2023. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3836–3847.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*.

Zhu, Z.; Huang, G.; Deng, J.; Ye, Y.; Huang, J.; Chen, X.; Zhu, J.; Yang, T.; Lu, J.; Du, D.; et al. 2021. Webface260m: A benchmark unveiling the power of million-scale deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10492–10502.