DynFaceRestore: Balancing Fidelity and Quality in Diffusion-Guided Blind Face Restoration with Dynamic Blur-Level Mapping and Guidance

Supplementary Material

1. Implementation Details

1.1. Dynamic Blur-Level Mapping

Architecture: With the objective of transforming the degraded input y into a Gaussian-blurred counterpart $\acute{y}$, the proposed Dynamic Blur-Level Mapping (DBLM) is composed of two primary modules: the Standard Estimation (SE) module and the Restoration Model (RM). Specifically, the SE predicts the Gaussian blur standard deviation $\hat{std}^*$ corresponding to y, while the RM refines y into the high-quality (HQ) distribution. The final Gaussian-blurred output, as described in Eq. (8) of the main paper, is rewritten as follows:

$$\acute{y} = k^{\hat{std}^*} \otimes RM(y), \tag{1}$$

where RM represents any SOTA pre-trained restoration model capable of mapping degraded inputs directly to high-quality (HQ) outputs. In our experiments, we adopt SwinIR [7] architecture, which has been well-trained in DifFace [17]. k^{std^*} denotes as Gaussian blur kernel defined by $\hat{std}^*$ estimated by SE module.

As shown in Fig. 1, the SE comprises a Transfer Model (TM) and a Standard Deviation Estimator (SDE). The TM, built on the SwinIR architecture [7], converts the degraded input y into an intermediate Gaussian-blurred image $\hat{y}'$. Subsequently, the SDE processes this intermediate image to estimate the corresponding Gaussian blur level $\hat{std}^*$. The detailed architecture of the SDE is provided in Tab. 1.

Data Preparation for Training SE: To train the SE module and implement the concept of DBLM, we first prepare the training labels. Specifically, for synthesized low-quality (LQ) facial images, we collect pairs of ground truth labels: (1) kernel standard deviations std^* to supervise the SDE and (2) Gaussian-blurred images $\tilde{x}$, generated by degrading the high-quality (HQ) image x using k^{std^*}, to supervise the TM. The synthesized low-quality (LQ) facial images y are created using the degradation pipeline described below:

$$y = \{JPEG_\delta[(x \otimes k_\sigma) \downarrow_C + n_\zeta]\} \uparrow_C . \tag{2}$$

Here, we use the same hyperparameter settings as [10] for synthesizing low-quality (LQ) facial images. Additionally, the kernel standard deviation label std^* is determined by using Eq. (3), and the corresponding Gaussian-blurred ground truth is generated using $\tilde{x} \equiv k_y^{std^*} \otimes x$.

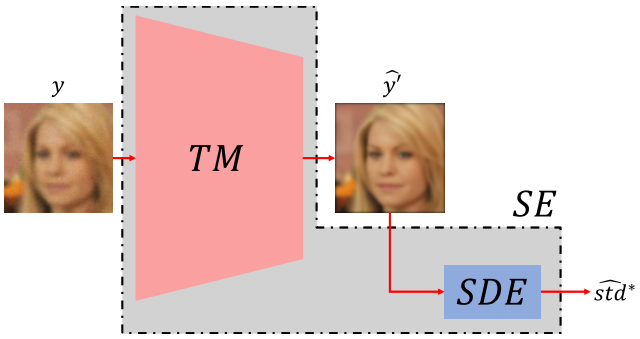

Figure 1. The SE consists of TM and a SDE, predicting the Gaussian blur level corresponding to the degraded input based on intermediate Gaussian blur image $\hat{y}'$.

$$\begin{aligned} std^* &\equiv \underset{std \in [std_{min}, std_{max}]}{argmin} (std), \\ s.t. \quad & \| k^{std} \otimes RM(y) - k^{std} \otimes x \|^1 < \xi. \end{aligned} \tag{3}$$

Here, ξ represents the error tolerance, x denotes the high-quality (HQ) ground truth, and $\otimes$ signifies the convolution operation. The search space for the standard deviation range, $[std_{min}, std_{max}]$, is defined between 0.1 and 15.0 with an interval of 0.1. To solve the optimization problem in Eq. (3), we employ a brute-force method.

Training procedure of SE: The training procedure of SE begins with SDE within the SE. Here, Gaussian-blurred images are generated by Eq. (4) and used to train the SDE by L_{SDE} defined in Eq. (5).

$$\ddot{y} = k^{\ddot{std}} \otimes x. \tag{4}$$

L_{SDE} denotes the loss function that aligns the SDE's predictions with the actual blur levels $\ddot{std}$. After completing this training phase, the SDE reliably estimates the Gaussian kernel for any Gaussian-blurred image, forming a robust foundation for effective blur-level prediction in the subsequent stages of the DBLM framework.

$$L_{SDE} = \mathbb{D}(SDE(\ddot{y}), \ddot{std}). \tag{5}$$

Subsequently, we train the whole SE in an end-to-end matter, where the SDE and TM modules in the SE are trained using the following loss:

$$L_{SE} = \mathbb{D}(\hat{std}^*, std^*) + \gamma_{std} \mathbb{D}(\hat{y}', \tilde{x}), \tag{6}$$

Table 1. Architecture of SDE.

architecture	channels
Conv2d: kernel size: 3×3; stride: 1	3 → 64
BatchNormalize2d	64
LeakyReLU	-
Conv2d: kernel size: 3×3; stride: 1	64 → 64
BatchNormalize2d	64
LeakyReLU	-
Conv2d: kernel size: 3×3; stride: 2	64 → 128
BatchNormalize2d	128
LeakyReLU	-
Conv2d: kernel size: 3×3; stride: 1	128 → 128
BatchNormalize2d	128
LeakyReLU	-
Conv2d: kernel size: 3×3; stride: 2	128 → 256
BatchNormalize2d	256
LeakyReLU	-
Conv2d: kernel size: 3×3; stride: 1	256 → 256
BatchNormalize2d	64
LeakyReLU	-
AvgPool2d	-
Linear	256 → 256
LeakyReLU	-
Linear	256 → 256
Linear	256 → 1

where $\mathbb{D}$ is the L1 distance, γ_{std} is a weighting factor for balancing, $\hat{std}^*$ is the output of SDE, $\hat{y}'$ is the output of TM, and $\tilde{x} \equiv k_y^{std^*} \otimes x$. This allows the SE to accurately predict the Gaussian blur level corresponding to any degraded input y.

Finally, as described in the main paper, the SE and RM are integrated to form the proposed DBLM, transforming the unknown degraded input y into its corresponding Gaussian-blurred version $\acute{y}$. This transformation simplifies the Blind Face Restoration task into a Gaussian deblurring problem.

1.2. Dynamic Starting Step Look-up Table

As mentioned in Sec.4.2 in our main paper, the optimal starting timestep t_{std} for each standard deviation is defined as follows:

$$t_{std} = \underset{t}{argmin} \quad (log(\mathbf{X}_t) - log(\tilde{\mathbf{Y}}_t^{std}) \leq tol), \qquad (7)$$

where tol represents the maximum tolerance, $\mathbf{X}_t$ and $\tilde{\mathbf{Y}}_t^{std}$ denote the expected values of x_t and $\tilde{y}_t^{std}$ in a training set. We set tol to 1×10^{-3} to construct the Dynamic Starting Step Look-up Table (DSST), which pairs each standard deviation std with its corresponding starting step t_{std}. Given the Gaussian-blur image $\acute{y}$, the estimated standard deviation $\hat{std}^*$ from SE is equalized and serves as a key to retrieve the corresponding starting step from the DSST.

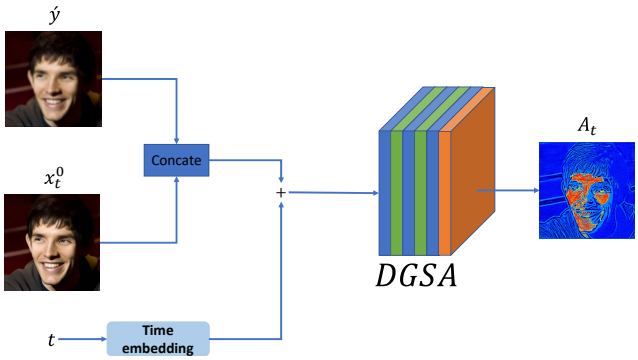

Figure 2. The inference flow of the DGSA is as follows: the measurement $\acute{y}$ and the high-quality prediction x_t^0 from x_t are first concatenated to form a combined input. This concatenated input is then integrated with the current timestep t, which is processed through the time embedding module of the diffusion model. The resulting features are input into DGSA, generating a region-specific guidance scale map that dynamically adjusts the guidance scale at each timestep to balance fidelity and detail preservation.

Table 2. Network architecture of DGSA.

architecture	channels
Conv2d: kernel size: 3×3	6 → 64
ELU	-
Conv2d: kernel size: 3×3	64 → 64
ELU	-
Conv2d: kernel size: 3×3	64 → 3
ReLU-1	-

1.3. Dynamic Guidance Scale Adjuster

Network Architecture: The architecture of the Dynamic Guidance Scale Adjuster (DGSA) is outlined in Tab. 2. DGSA consists of three convolutional layers, with ELU activation functions applied after the first two layers. The final layer uses the ReLU-1 activation function to constrain the output within the range of 0 to 1. At timestep t, the inputs to DGSA include the degraded measurement $\acute{y}$, the high-quality (HQ) prediction x_t^0 derived from x_t, and the current timestep t.

In the implementation of DGSA, as illustrated in Fig. 2, $\acute{y}$ and x_t^0 are concatenated before being input into the DGSA. This design ensures that the model effectively captures low-frequency information from both inputs to preserve fidelity. The timestep t is then embedded and combined with the concatenated features to determine the localized diffusion power required at each region and timestep. The output of DGSA is a pixel-wise guidance scale map with values ranging from 0 to 1, matching the image dimensions. Higher values indicate stronger guidance influence to preserve fidelity, while lower values relax the guidance to utilize the DM's realistic facial generation capabilities. This behavior is visualized in Fig. 5.

Algorithm 1 DGSA training

Require: y: Unknown degraded LQ input; $iter_{total}$: Total training iterations; γ_i: Weight factor of SWT four subbands;

$iter$ = 0;
while $iter < iter_{total}$ **do**
  $\acute{y}, \hat{std^*} = DBLM(y), SE(y)$;
  $t_{start} = DSST(\hat{std^*})$;
  $t \sim Uniform(0, t_{start})$;
  $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \epsilon \sim N(0,1)$;
  $x_t^0 = \frac{1}{\sqrt{\bar{\alpha}_t}}x_t - \sqrt{\frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}}\epsilon_\theta$;
  compute Gaussian blur kernel k using $\hat{std^*}$;
  $x'_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta) + \sigma_t\epsilon, \epsilon \sim N(0,1)$;
  $A_t = DGSA(\acute{y}, x_t^0, t)$
  $x_{t-1} = x'_{t-1} - A_t \times \nabla_{x_t} \left\| \acute{y} - k_t \otimes x_t^0 \right\|^2$;
  $x_{t-1}^0 = \frac{1}{\sqrt{\bar{\alpha}_{t-1}}}x_{t-1} - \sqrt{\frac{1-\bar{\alpha}_{t-1}}{\bar{\alpha}_{t-1}}}\epsilon_\theta$;
  $L_{DGSE} = \sum_i \gamma_i \mathbb{D}(SWT(x_{t-1}^0)_i, SWT(x_0)_i)$
  $+ \quad DISTS(x_{t-1}^0, x_0)$;
  update DGSA using L_{DGSE};
  $iter = iter + 1$;
end while

Training procedure: The DGSA training process is divided into two stages for optimal performance. Initially, DGSA is trained using actual Gaussian blurry images $\ddot{y}$ as the measurement inputs, generated based on Eq. (4). That is, we replace the output of DBLM $\acute{y}$ with $\ddot{y}$ in this stage. This stage consists of 20,000 iterations, allowing DGSA to learn robust guidance scale mappings for well-defined blurry images as its measurement input. In the second stage, DGSA undergoes fine-tuning with its original input, $\acute{y}$, representing the measurement predicted by our DBLM. This fine-tuning phase, spanning 7,000 iterations, further refines DGSA's ability to adapt to real-world degraded inputs. During training, the DBLM module is kept frozen, and the DSST is pre-established.

We train DGSA at each randomly sampled timestep using the following loss function:

$$L_{DGSA} = \sum_i \gamma_i \mathbb{D}(SWT(x_{t-1}^0)_i, SWT(x_0)_i) + DISTS(x_{t-1}^0, x_0), \tag{8}$$

where γ_i are the weighted factors of the four subbands (LL, LH, HL, HH) decomposed by Stationary Wavelet Transformation (SWT) [3, 5], x_0 is the learning target, and x_{t-1}^0 is the HQ prediction based on x_{t-1} at timestep $t-1$. Here, $\mathbb{D}$ and $DISTS$ [1, 5] are the L1 reconstruction loss and the perceptual loss, respectively.

The L1 reconstruction loss is applied across four subbands obtained via Stationary Wavelet Transform (SWT), with weighted factors γ_i for LL, LH, HL, and HH set to 0.00, 0.01, 0.01, and 0.05, respectively. These weights prioritize high-frequency details, ensuring that the details of the samples adjusted by guidance remain consistent with the ground truth. Furthermore, we utilize DISTS to maintain the perceptual quality of the output. The complete training procedure is detailed in Algorithm 1.

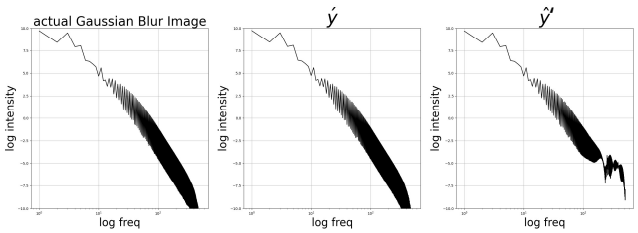

Figure 3. The frequency responses of actual Gaussian blurry images, $\acute{y}$, and $\hat{y}'$. Here, $\hat{y}'$, the output of TM (refer to Fig. 1), exhibits significant differences in the high-frequency components compared to actual Gaussian-blurred images. In contrast, $\acute{y}$, estimated using Eq. (1), closely matches the frequency response of the actual Gaussian-blurred image.

1.4. Inference

Our DynFaceRestore framework is designed to be easily extendable, allowing for the use of multiple guidance sources. This flexibility enables a balance between perceptual quality and fidelity. In all our experiments, we utilize guidance from three blurred images, as outlined in Algorithm 2, with the weights $\lambda^{i\in[1,2,3]}$ are set to 0.7, 0.2, and 0.1, respectively. Further experimental analysis and discussions regarding the multiple guidance setup are provided in Sec. 2.3.

2. Extra Experiments and Ablation Studies

2.1. Kernel Mismatch

The difference between $\hat{y}'$ and $\acute{y}$: The analysis in Fig. 3 provides a detailed comparison between $\hat{y}'$ and $\acute{y}$, offering a strong rationale for designing the DBLM output as $\acute{y}$ rather than directly using the output of TM, $\hat{y}'$. By transforming $\hat{y}'$ and $\acute{y}$ into the frequency domain, a significant disparity in high-frequency intensity emerges when comparing $\hat{y}'$ to actual Gaussian-blurred images.

In contrast, explicitly convolving the output of $RM(y)$ with a Gaussian kernel to produce $\acute{y}$ yields high-frequency intensity closely aligned with that of actual Gaussian-blurred images. This alignment suggests that the high-frequency discrepancy in $\hat{y}'$ is a crucial contributor to kernel mismatch, adversely affecting restoration fidelity. Furthermore, the quantitative results in Tab. 3 confirm these findings, showing that using $\acute{y}$ as diffusion guidance in DBLM achieves superior fidelity over $\hat{y}'$, further validating the effectiveness of explicitly constructing $\acute{y}$.

Refinement of the Standard Deviation: In Fig. 4, experimental evidence highlights the critical role of kernel re-

Algorithm 2 Inference - Multiple Guidance

Require: y: Unknown degraded LQ input; $\lambda^{i\in[1,2,3]}$: weights of each guidance;
Output: x_0: HQ sampled image;

$\acute{y}, \hat{std^*} = DBLM(y), SE(y)$;
$\hat{std^*}^{i\in[1,2,3]} = [\hat{std^*}, \hat{std^*} - 1, \hat{std^*} - 2]$
compute Gaussian blur kernel $k^{i\in[1,2,3]}$ using $\hat{std^*}^{i\in[1,2,3]}$;
$\acute{y}^{i\in[1,2,3]} = [\acute{y}, k^2 \otimes RM(y), k^3 \otimes RM(y)]$
$t_{start}^{i\in[1,2,3]} = DSST(\hat{std^*}^{i\in[1,2,3]})$;
$std_{t_{start}}^{i\in[1,2,3]} = \hat{std^*}^{i\in[1,2,3]}$;
$x_{t_{start}} = \sqrt{\bar{\alpha}_{t^1_{start}}}\acute{y}^1 + \sqrt{1-\bar{\alpha}_{t^1_{start}}}\epsilon, \epsilon \sim N(0,1)$;
for $t = t^1_{start} \cdots 1$ **do**
    $x_t^0 = \frac{1}{\sqrt{\bar{\alpha}_t}} x_t - \sqrt{\frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}}\epsilon_\theta$;
    $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta) + \sigma_t\epsilon, \epsilon \sim N(0,1)$;
    compute Gaussian blur kernel $k_t^{i\in[1,2,3]}$ using $std_t^{i\in[1,2,3]}$;
    $A_t = DGSA(\acute{y}^1, x_t^0, t)$;
    if $t \in [t^1_{start}, t^2_{start}]$ **then**
        $x_{t-1} = x'_{t-1} - A_t \times \nabla_{x_t} \left\|\acute{y}^1 - k_t^1 \otimes x_t^0\right\|^2$;
        $std^1_{t-1} = std^1_t - \sqrt{\bar{\alpha}_t} \times \nabla_{k_t^1} \left\|\acute{y}^1 - k_t^1 \otimes x_t^0\right\|^2$;
    end if
    if $t \in [t^2_{start}, t^3_{start}]$ **then**
        $x_{t-1} = x'_{t-1} - A_t \times \nabla_{x_t} \sum_{i=1}^{2}(\frac{\lambda^i}{\lambda^1+\lambda^2}) \left\|\acute{y}^i - k_t^i \otimes x_t^0\right\|^2$;
        $std_{t-1}^{i\in[1,2]} = std_t^{i\in[1,2]} - \sqrt{\bar{\alpha}_t} \times \nabla_{k_t^{i\in[1,2]}}(\frac{\lambda^i}{\lambda^1+\lambda^2}) \left\|\acute{y}^{i\in[1,2]} - k_t^{i\in[1,2]} \otimes x_t^0\right\|^2$;
    end if
    if $t \in [t^3_{start}, 0]$ **then**
        $x_{t-1} = x'_{t-1} - A_t \times \nabla_{x_t} \sum_{i=1}^{3}(\frac{\lambda^i}{\lambda^1+\lambda^2+\lambda^3}) \left\|\acute{y}^i - k_t^i \otimes x_t^0\right\|^2$;
        $std_{t-1}^{i\in[1,2,3]} = std_t^{i\in[1,2,3]} - \sqrt{\bar{\alpha}_t} \times \nabla_{k_t^{i\in[1,2,3]}}(\frac{\lambda^i}{\lambda^1+\lambda^2+\lambda^3}) \left\|\acute{y}^{i\in[1,2,3]} - k_t^{i\in[1,2,3]} \otimes x_t^0\right\|^2$;
    end if
end for
return x_0

Table 3. Ablation study on different output types of DBLM in the CelebA-Test dataset. Here, $\hat{y}'$ and $\acute{y}$ represent different approximations of the Gaussian-blurred image $\tilde{x}$. The results clearly show that $\acute{y}$ provides a closer approximation of $\tilde{x}$, resulting in improved fidelity.

Gaussian Blur image	PSNR↑	SSIM↑	LPIPS↓	FID↓	IDA↓	LMD↓
$\hat{y}'$	24.261	0.661	0.332	**14.185**	0.756	3.428
$\acute{y}$	**24.349**	**0.664**	0.332	14.780	**0.748**	**3.419**

finement at each step in mitigating kernel mismatch during the sampling process. Using an actual Gaussian blur measurement with a kernel of $std = 3.0$ as the guidance, the experiment in Fig. 4 evaluates outcomes without employing DBLM, DSST, or DGSA to focus solely on analyzing the kernel refinement function. When the kernel std is not refined and fixed to the wrong standard deviation at each step, a substantial deviation between the final sampled results and the ground truth is evident, even with guidance applied. Notably, the results closely align with the ground truth only when the kernel std is accurately set to 3.0, as further validated by the "PSNR over std" analysis shown in Fig. 4.

In contrast, with kernel std refinement applied at each step, the sampled results exhibit remarkable fidelity to the ground truth, even when the initial kernel std estimation deviates from the actual value. This outcome underscores the robustness of the kernel refinement strategy in addressing inaccuracies in initial kernel estimation, effectively bridging the gap caused by kernel mismatch. These findings validate the necessity of dynamic kernel std adjustments to achieve high-quality restoration.

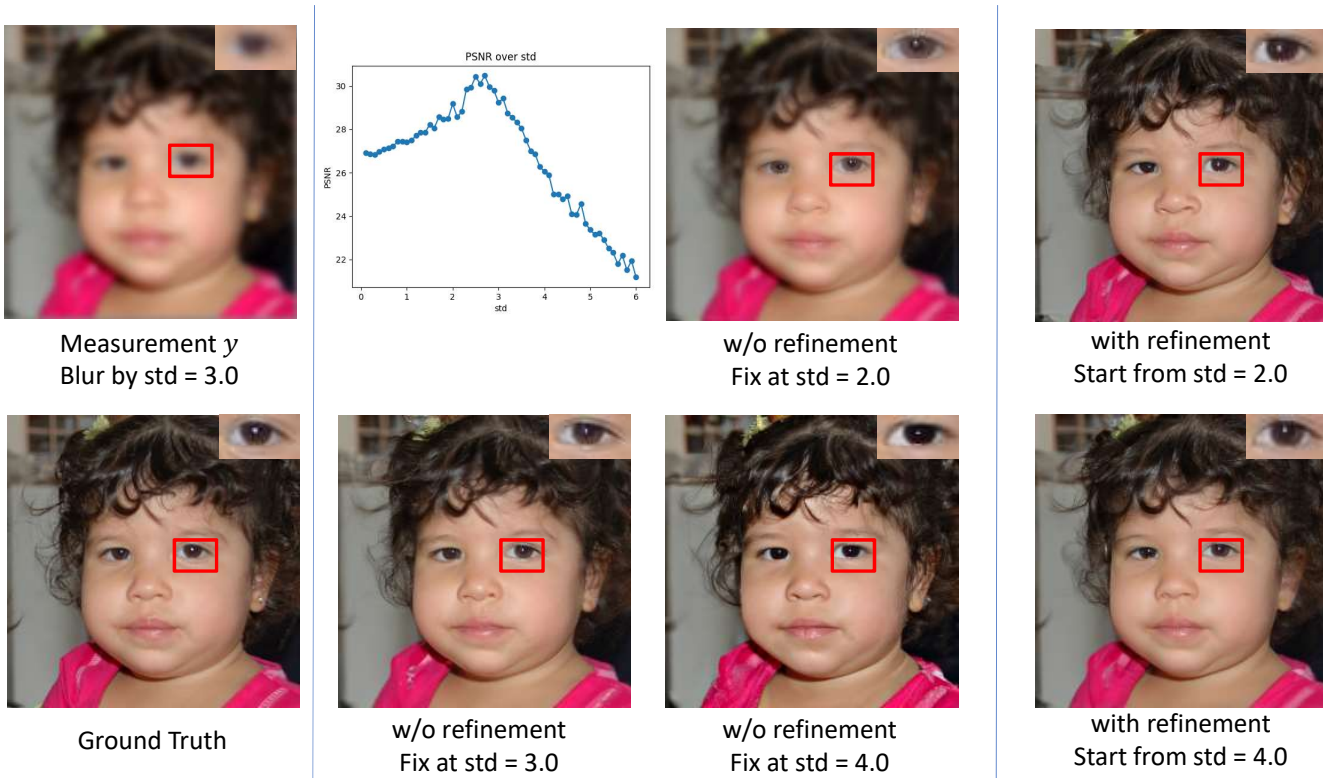

Figure 4. Comparison of results with and without kernel refinement is conducted using an actual Gaussian-blurred image with a kernel standard deviation (std) of 3.0 as the input measurement for guidance. During the diffusion sampling process, applying guidance adjustment without kernel refinement—e.g., fixing the kernel std to 2.0 or 4.0—leads to significant deviations from the ground truth due to kernel mismatch. When the fixed std is set to 2.0, the mismatch limits the diffusion model's ability to add details, resulting in overly blurry outputs that rely too heavily on the measurement. Conversely, with a fixed std of 4.0, the diffusion model overly enhances the image, introducing hallucinated artifacts. In contrast, incorporating kernel refinement alongside guidance adjustment enables the sampling process to correct initial inaccuracies in kernel prediction. Even when starting with an imperfect kernel std, such as 2.0 or 4.0, the refined approach ensures that the final outputs align much more closely with the ground truth. This demonstrates the robustness of the refinement strategy in mitigating kernel mismatch issues.

2.2. Visualize of DGSA

The output of DGSA is visualized in Fig. 5 to highlight its effectiveness. In each diffusion timestep, DGSA dynamically adjusts the guidance scale region-wise. The guidance scale gradually decreases for detail-rich areas such as hair and wrinkles, enabling these regions to harness more of the pre-trained DM's high-quality image prior to sampling, resulting in realistic and refined facial details. Conversely, the guidance scale remains elevated for structural regions, such as the eyes and mouth, to preserve their geometric integrity and ensure the reconstructed image aligns closely with the ground truth. This adaptive balance between fidelity and detail generation leads to high realism outputs while maintaining structural consistency.

2.3. Multiple Guidance

In this section, we evaluate the impact of varying the number of guidance sources. To ensure accurate assessment, DGSA is excluded to eliminate any potential external influences in this experiment. Each guidance source corresponds to a specific standard deviation, which changes following a defined pattern. For example, with n guidance sources, we have $\acute{y}'^{i\in[1,2,\dots,n]}$ corresponding to the standard deviations $\hat{std}^*, \hat{std}^* - 1, \dots, \hat{std}^* - (n-1)$ and the weights $\lambda^{i\in[1,2,\dots,n]}$.

Since $\acute{y}'^1$ with $\hat{std}^*$ retains more reliable low-frequency structural information, λ^1 is assigned a higher weight. In contrast, smaller standard deviations values $\acute{y}'^{i\in[2,3,4]}$ provide higher-frequency details with reduced confidence, and thus, $\lambda^{i\in[2,3,4]}$ are assigned lower weights, as outlined by λ^i in Tab. 4.

As shown in Tab. 4, increasing the number of guidance sources amplifies the influence of the restoration model (RM), resulting in improved fidelity and higher scores in PSNR. However, perceptual quality (FID) is optimized when using a single guidance source, which allows more

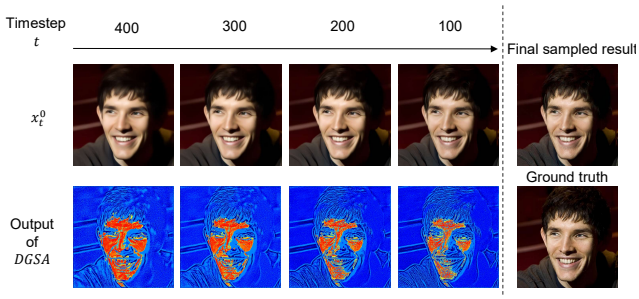

Figure 5. Visualization of DGSA at various timesteps during the sampling process. Here, x_t^0 is the HQ prediction of x_t at timestep t. DGSA generates guidance maps, where blue regions indicate less guidance and rely more on the diffusion model (DM) to add details, while red regions signify stronger guidance with less reliance on the DM. As t decreases, areas requiring the DM's HQ prior, such as hair, display a lower guidance scale to enhance detail. Conversely, facial regions maintain a higher guidance scale to ensure fidelity preservation.

Table 4. Ablations of the different numbers of guidance in CelebA-Test dataset. # denotes the numbers of guidance and λ^i are the weights of each guidance. The best performance is highlighted with **bold**.

#	λ^i	PSNR↑	IDA↓	FID↓
1	[1.0]	25.014	0.7242	**18.452**
2	[0.8, 0.2]	25.049	0.7238	18.98
3	[0.7, 0.2, 0.1]	25.107	**0.7236**	19.786
4	[0.7, 0.1, 0.1, 0.1]	**25.185**	0.7242	20.947

reliance on the diffusion model. This demonstrates the flexibility of our framework, enabling users to tune the desired output by controlling the number of guidance sources. Consequently, our framework effectively achieves a balance between perceptual quality and fidelity.

2.4. Different Starting Steps

By setting up the output of DBLM as a Gaussian-blurred image based on the input degradation severity, we can leverage this property to identify the optimal timestep (t^*), as outlined in Sec. 4.2 using "Dynamic Starting Step Lookup Table". Note that t^* is automatically determined for different datasets. To demonstrate the effectiveness of our approach, we conduct experiments with different starting timesteps, as shown in Tab. 5. Larger timesteps ($> t^*$) result in a loss of information from the guidance observation, leading to reduced fidelity. Conversely, smaller timesteps ($< t^*$) provide insufficient iterations to recover fine details, thereby diminishing the quality of the reconstruction.

2.5. Different SOTA Restoration Model

We evaluate the impact of different Restoration Models (RMs) on overall performance by replacing our pre-trained RM with various alternatives, including GAN-based GFP-GAN [6], CodeBook-based RestoreFormer [13] and deterministic-based SwinIR [7]. To ensure a fair comparison, we first fine-tune each RM on the FFHQ [4] dataset before integrating it into our framework.

Table 5. Ablation study of same blurred image with different starting steps in CelebA-Test. Top performances in **bold** and underline.

Steps	PSNR↑	SSIM↑	LPIPS↓	FID↓	IDA↓	LMD↓
400 ($< t^*$)	23.393	0.659	0.394	42.055	0.853	3.781
1000 ($> t^*$)	24.172	0.657	0.336	**14.556**	0.761	3.470
t^* [690,925]	**24.349**	**0.664**	**0.332**	14.780	**0.748**	**3.419**

As shown in Tab. 6 and visualized in Fig. 6, GFP-GAN suffers from poor realism (higher FID). Integrating our approach significantly improves the realism of GFP-GAN while maintaining competitive fidelity. Also, we find that RestoreFormer outperforms GFP-GAN in realism, and our method can further enhance RestoreFormer in terms of fidelity, structural integrity, PSNR and SSIM. Finally, SwinIR achieves the highest PSNR and SSIM but produces overly smooth images, compromising realism. Incorporating our method with SwinIR substantially reduces FID, balancing fidelity and realism. Overall, our method consistently improves realism across different RMs while maintaining competitive fidelity, achieving a more optimal trade-off between perceptual quality and fidelity.

3. More Visualization

This section presents additional qualitative comparisons with state-of-the-art methods, including GPEN [16], GFPGAN [6], RestoreFormer [13], CodeFormer [18], DiffBIR [8], DAEFR [12], PGDiff [15], DifFace [17], DR2 [14]and 3Diffusion [9].

For the CelebA-Test dataset, Fig. 7 demonstrates that our DynFaceRestore not only produces reconstructions closer to the ground truth than competing methods but also offers superior perceptual quality. This highlights the balance between fidelity and perceptual quality in our approach.

For the three real-world datasets—LFW-Test [2], Wider-Test [18], and Webphoto-Test [6]—the results in Fig. 8, Fig. 9, and Fig. 10, respectively, highlight the advantages of DynFaceRestore. Specifically, our approach consistently generates more realistic facial details, such as hair strands and beards, while effectively preserving the global structure and local textures.

4. Limitations

A major limitation of our proposed method is its computational complexity, as shown in Tab. 1 (main paper). This is primarily due to the optimization of kernel prediction and the incorporation of DM guidance at each step. The acceleration through DDIM [11] has yet to be explored. Therefore,

Table 6. Differnet RM model comparisons to in CelebA-Test. The best and second performances are highlighted with **bold** and underline.

Type	Method	PSNR↑	SSIM↑	LPIPS↓	FID↓	IDA↓	LMD↓
GAN	GFP-GAN	22.841	0.620	0.355	23.860	0.822	4.793
	Ours + GFP-GAN	**23.867**	**0.649**	**0.340**	**15.943**	**0.772**	**3.654**
CodeBook	RestoreFormer	23.001	0.592	0.376	22.874	0.783	4.464
	Ours + RestoreFormer	**23.794**	**0.655**	**0.341**	**17.042**	**0.820**	**3.859**
Deterministic	SwinIR	**26.177**	**0.746**	0.377	61.209	**0.720**	**3.101**
	Ours + SwinIR	24.349	0.664	**0.332**	**14.78**	0.748	3.419

reducing the denoising steps and selectively applying guidance to the critical step offers a promising research direction to address this challenge.

Additionally, as illustrated in Fig. 11, the DBLM module in our DynFaceRestore framework exhibits limitations when handling the complex degradation patterns observed in old photographs, such as incomplete restoration due to residual artifacts (Fig. 11 left) or geometric distortions (Fig. 11 right). These limitations result in difficulties during the subsequent DM sampling and guidance processes, leading to difficulties in accurately identifying regions that require fidelity preservation and refinement. As a result, the overall restoration quality is degraded. We attribute this issue to the training of the pre-trained restoration model, which fails to capture the diverse and severe degradations commonly observed in old photographs. A straightforward and effective solution to overcome these limitations is to replace the RM with more advanced restoration models.

Moreover, real-world images often exhibit spatially varying degradations, posing a significant challenge in perfectly addressing the kernel mismatch issue. Treating the entire image as uniformly degraded can lead to undesired artifacts, as shown in Fig. 12, which our proposed method has not yet effectively resolved. Addressing this issue is crucial for future research, and a direct approach is to divide regions and estimate degradations separately.

References

[1] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE transactions on pattern analysis and machine intelligence*, 44(5):2567–2581, 2020. 3

[2] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008. 6

[3] Björn Jawerth and Wim Sweldens. An overview of wavelet based multiresolution analyses. *SIAM review*, 36(3):377–412, 1994. 3

[4] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 6

[5] Cansu Korkmaz, A Murat Tekalp, and Zafer Dogan. Training generative image super-resolution models by wavelet-domain losses enables better control of artifacts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5926–5936, 2024. 3

[6] Ayushi Kumar and Avimanyou Vatsa. Influence of gfp gan on melanoma classification. In *2022 IEEE Integrated STEM Education Conference (ISEC)*, pages 334–339. IEEE, 2022. 6

[7] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1833–1844, 2021. 1, 6

[8] Xinqi Lin, Jingwen He, Ziyan Chen, Zhaoyang Lyu, Bo Dai, Fanghua Yu, Yu Qiao, Wanli Ouyang, and Chao Dong. Diffbir: Toward blind image restoration with generative diffusion prior. In *European Conference on Computer Vision*, pages 430–448. Springer, 2024. 6

[9] Xiaobin Lu, Xiaobin Hu, Jun Luo, Ben Zhu, Yaping Ruan, and Wenqi Ren. 3d priors-guided diffusion for blind face restoration. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 1829–1838, 2024. 6

[10] Yunqi Miao, Jiankang Deng, and Jungong Han. Waveface: Authentic face restoration with efficient frequency recovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6583–6592, 2024. 1

[11] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 6

[12] Yu-Ju Tsai, Yu-Lun Liu, Lu Qi, Kelvin C. K. Chan, and Ming-Hsuan Yang. Dual associated encoder for face restoration, 2024. 6

[13] Zhouxia Wang, Jiawei Zhang, Runjian Chen, Wenping Wang, and Ping Luo. Restoreformer: High-quality blind face restoration from undegraded key-value pairs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17512–17521, 2022. 6

[14] Zhixin Wang, Ziying Zhang, Xiaoyun Zhang, Huangjie Zheng, Mingyuan Zhou, Ya Zhang, and Yanfeng Wang. Dr2: Diffusion-based robust degradation remover for blind face restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1704–1713, 2023. 6

[15] Peiqing Yang, Shangchen Zhou, Qingyi Tao, and Chen Change Loy. Pgdiff: Guiding diffusion models

for versatile face restoration via partial guidance. *Advances in Neural Information Processing Systems*, 36, 2024. 6

[16] Tao Yang, Peiran Ren, Xuansong Xie, and Lei Zhang. Gan prior embedded network for blind face restoration in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 672–681, 2021. 6

[17] Zongsheng Yue and Chen Change Loy. Diface: Blind face restoration with diffused error contraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1, 6

[18] Shangchen Zhou, Kelvin Chan, Chongyi Li, and Chen Change Loy. Towards robust blind face restoration with codebook lookup transformer. *Advances in Neural Information Processing Systems*, 35:30599–30611, 2022. 6

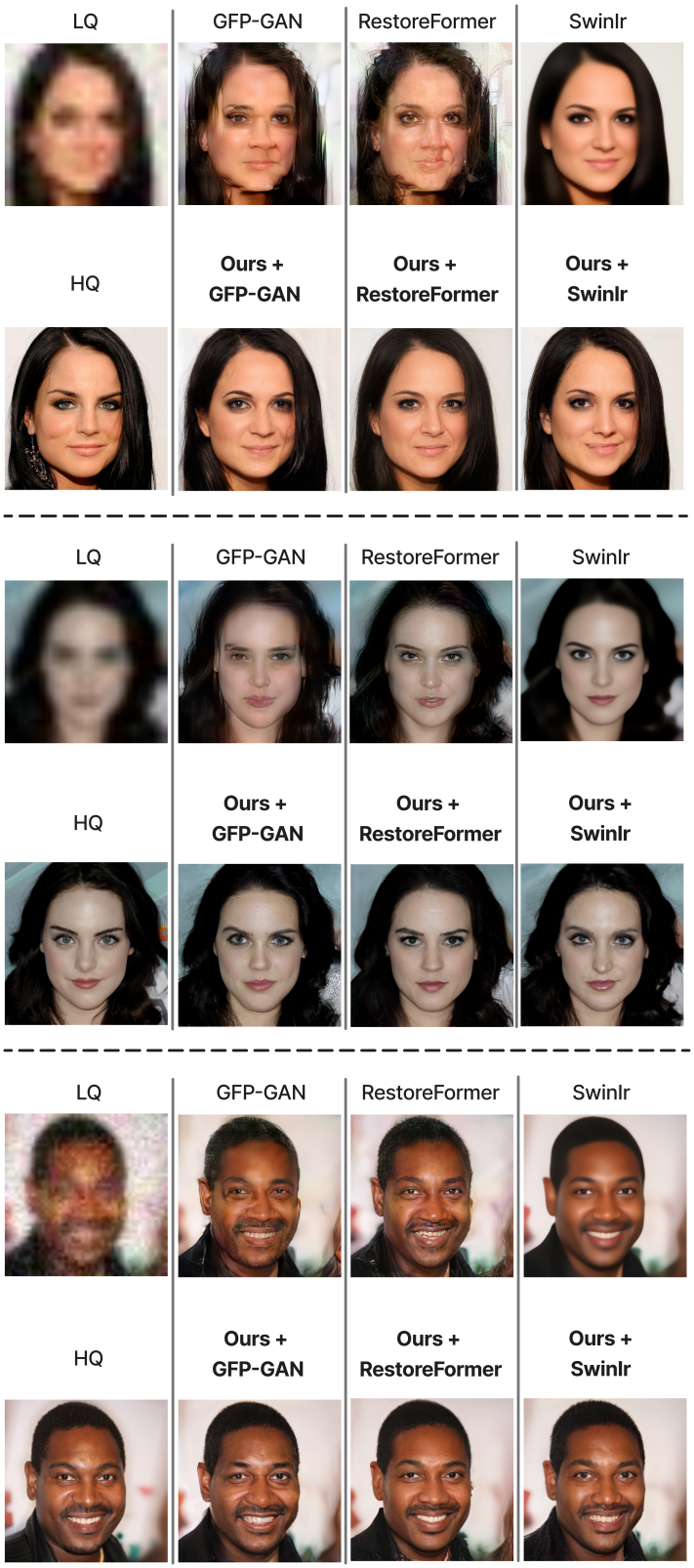

Figure 6. Ablation study with different RMs.

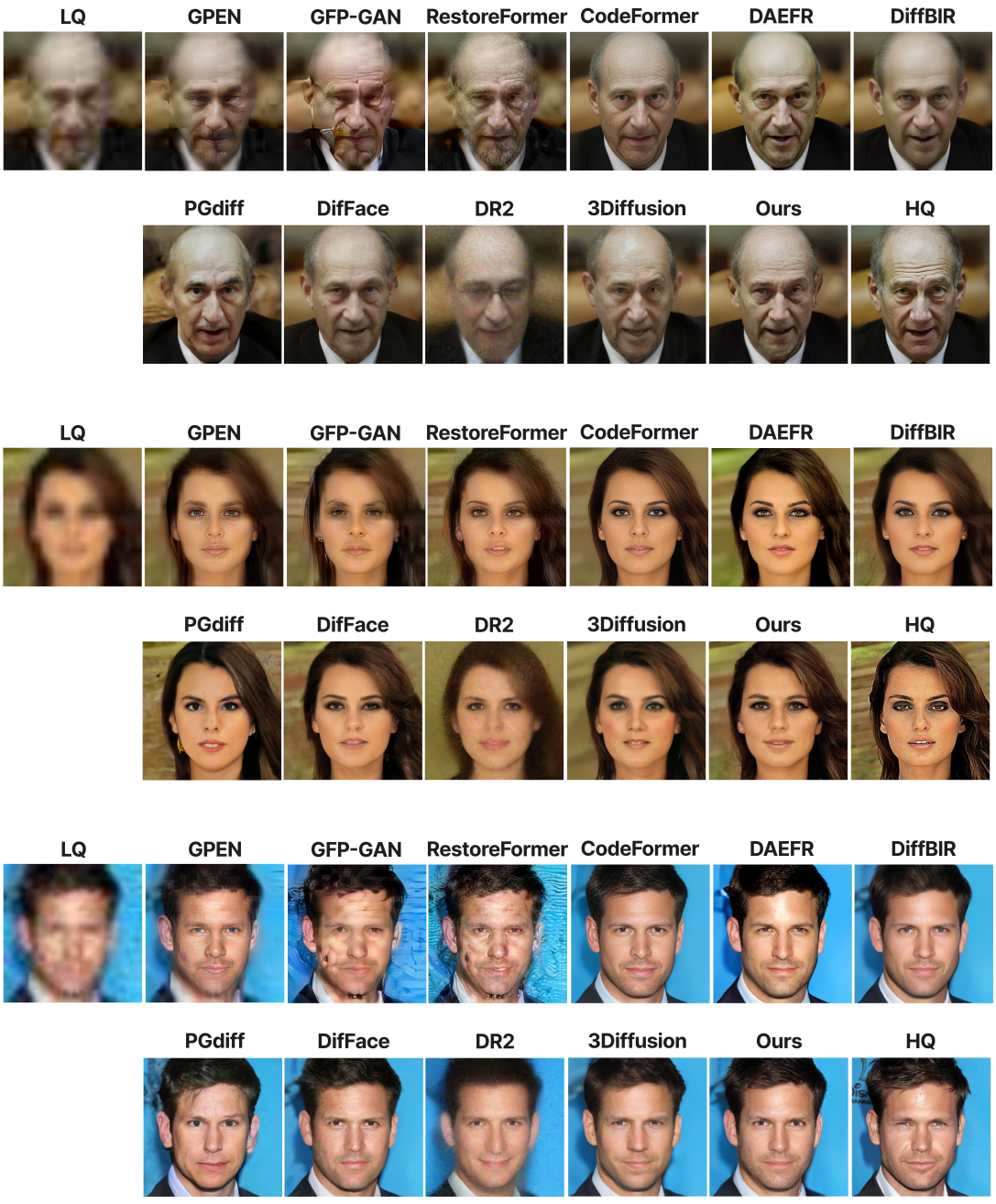

Figure 7. More visual comparisons on CelebA-Test. Our method achieves high-fidelity reconstructions while preserving natural facial features.

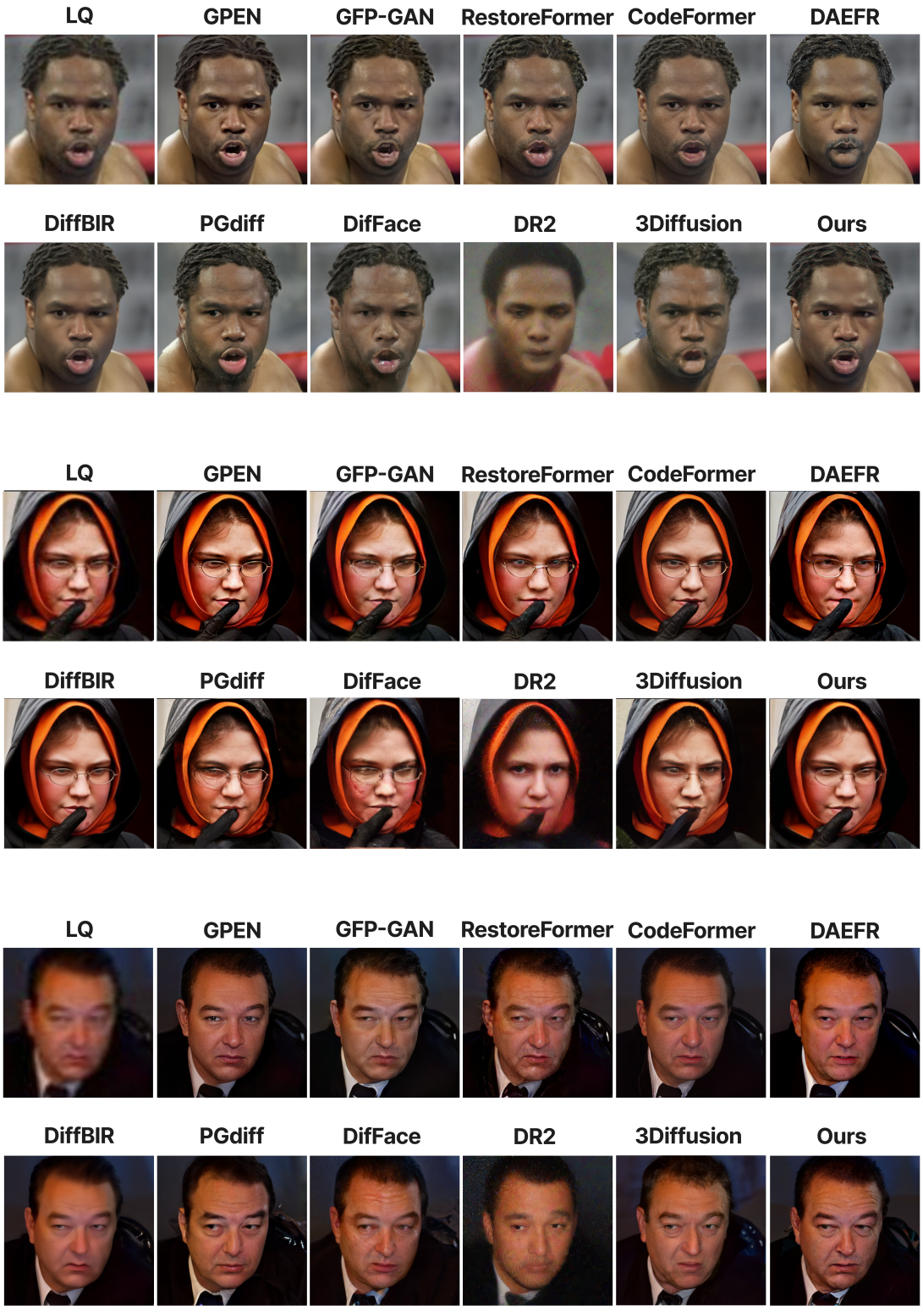

Figure 8. Qualitative results from LFW-Test demonstrate that our restoration method produces more natural features (e.g., eyes) and realistic details (e.g., hair) compared to other approaches, with improved fidelity.

Figure 9. More visual comparisons on Wider-Test. Our restoration method produces more natural features (e.g., eyes) and realistic details (e.g., hair, skin) compared to other approaches, with improved fidelity.

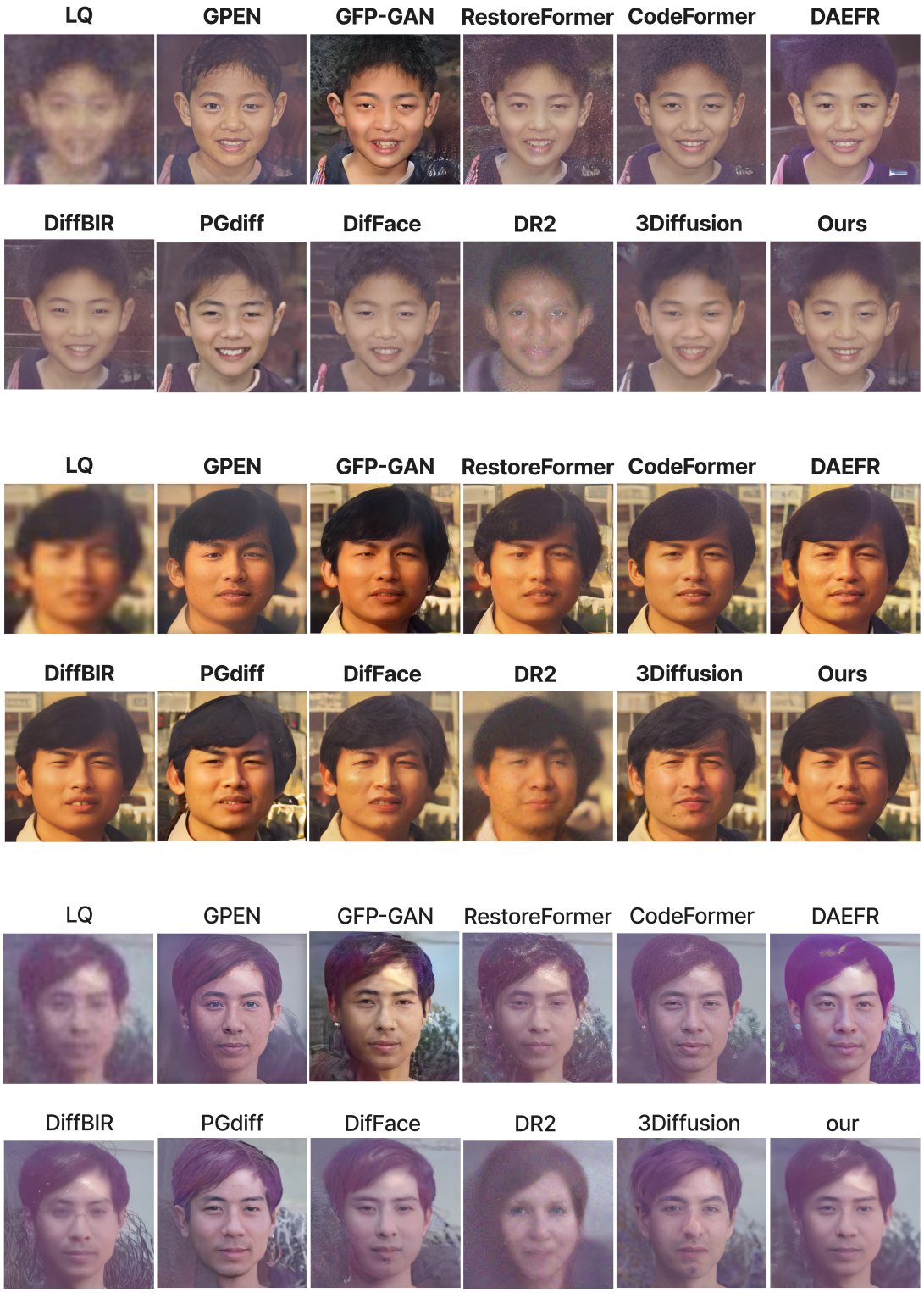

Figure 10. More visual comparisons on Webphoto-Test. Our restoration method produces more natural features (e.g., eyes) and realistic details (e.g., hair, skin) compared to other approaches, with improved fidelity.

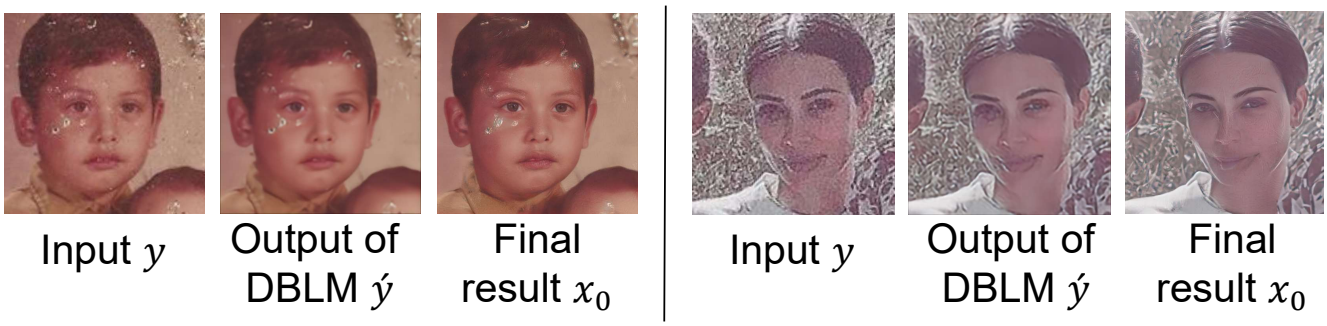

Figure 11. Limitations of Our DynFaceRestore. This issue likely arises from limitations in the degradation pipeline used to synthesize low-quality images for simulating real-world degradation. The current pipeline inadequately captures old photographs' diverse and severe degradation characteristics. Revising the pipeline to represent these complexities better is essential for improving restoration performance in such challenging scenarios.

Figure 12. Limitations of Our DynFaceRestore. We degrade images using Eq. (2), applying both global degradation (uniformly across the entire image) and local degradation (independently for different small regions). The results indicate the appearance of artifacts in the mouth and eye regions.