Kalman-Inspired Feature Propagation for Video Face Super-Resolution - Supplementary Materials -

Ruicheng Feng, Chongyi Li, and Chen Change Loy

S-Lab, Nanyang Technological University, Singapore
{ruicheng002, ccloy}@ntu.edu.sg
lichongyi25@gmail.com

A Method Details

A.1 Detailed Architecture

Kalman Filter Network. Our Kalman Filter Network, as illustrated in Figure A1, adopts two distinct parameterized modules in implicitly estimating uncertainty and Kalman gain. Note that the dynamic model from $\hat{\boldsymbol{z}}_{t-1}^{+}$ to prior estimation $\hat{\boldsymbol{z}}_{t}^{-}$ is omitted for simplicity in the illustration. The uncertainty network implicitly estimates the uncertainty of shape $h \times w \times c$, and the Kalman gain network calculates the corresponding Kalman gain $\mathcal{K}_t$ of shape $h \times w$ for each code token. The Spatial-Temporal Attention (ST-Attn) takes the current observed latent code $\tilde{\boldsymbol{z}}_t$ as a query and attends to the combination of the first frame $\tilde{\boldsymbol{z}}_1$ and previous frame $\tilde{\boldsymbol{z}}_{t-1}$. Inspired by [13], the spatial-temporal attention also takes the latent code of the first frame $\tilde{\boldsymbol{z}}_1$ as input, which serves as an anchor prior to all temporal attention.

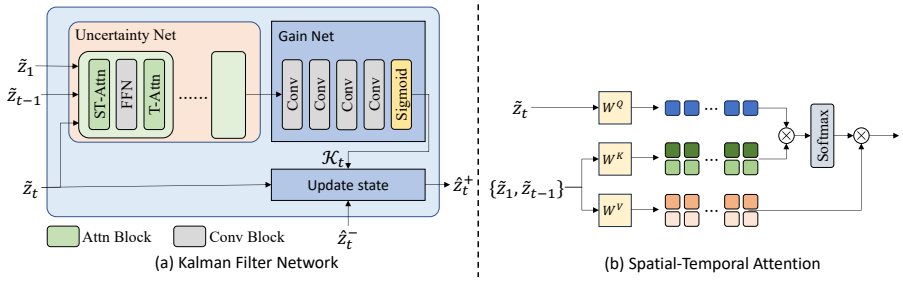

Fig. A1: Illustration of Kalman Filter Network. (a) We unfold and show one timestep of the Kalman filter network. The network mainly consists of two parametrization modules, *i.e.*, uncertainty network and the gain network. Here "ST-Attn" and "T-Attn" represent spatial-temporal attention and temporal attention, respectively. (b) The Spatial-Temporal Attention (ST-Attn) takes estimated observed latent code for current frame $\tilde{\boldsymbol{z}}_t$ as a query and attends to the combination of the first frame $\tilde{\boldsymbol{z}}_1$ and previous frame $\tilde{\boldsymbol{z}}_{t-1}$.

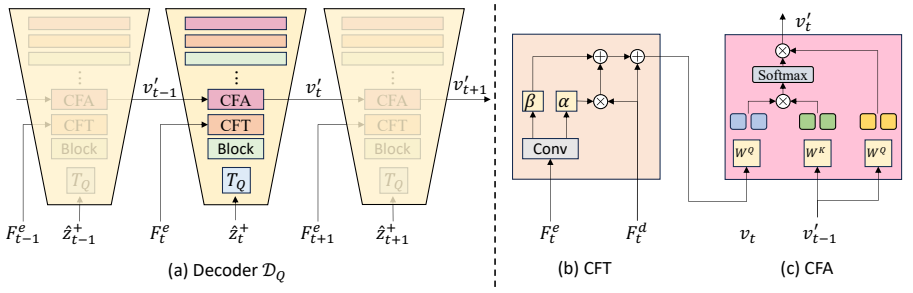

Fig. A2: Illustration of the integrated decoder. Controllable Feature Transformation (CFT) and Cross-Frame Attention (CFA). T_Q is a codebook lookup Transformer and quantization layer borrowed from CodeFormer [17]. Blocks are the basic conv blocks in the decoder. CFT is tailored for modulating the features of decoder F_d by the encoder's features F_e. CFA is adopted in the decoder to further promote local temporal consistency to regularize the information propagation.

Integrated Decoder. Figure A2 depicts how CFT and CFA layers are integrated into the decoder. Following [17], we leverage the encoder features to modulate the corresponding decoder features. Denoted F_e and F_d as the encoder and decoder features, respectively, the network learns an affine transformation defined by α and β.

$$v_t = F_d + (\alpha \cdot F_d + \beta), \tag{1}$$

where $\alpha, \beta = \mathcal{C}(F_e)$, and $\mathcal{C}$ is multiple convolution blocks. The CFT modules are adopted at multiple scales $16, 32, 64$, since shallow features of encoder would also bring forward corrupted information to the decoder and yield blurry results. This design facilitates fidelity reservation of each frame and hence improves temporal coherence.

To further enforce temporal information propagation and reduce local jitters, we adopt cross-frame attention modules, which search and match similar features from the previous frame and attend to them correspondingly. Specifically, given the latent features from the previous frame v_{t-1} and current frame v_t. They are projected onto the embedding space and output the features v'_i by $v'_i = \text{Attn}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d}}) \cdot V$, where

$$Q = W_Q \cdot v_t, K = W_K \cdot v_{t-1}, V = W_V \cdot v_{t-1}. \tag{2}$$

This module facilitates temporal information propagation in the decoder. We adopt CFA modules on features of small scale 16 and 32 to avoid introducing blur to the decoded results.

A.2 Algorithm Pseudocode

As shown in the below Algorithm , we present the pseudocode of our method. In this algorithm, we only show the process of inference.

Algorithm 1: Detailed algorithm of KEEP.

$\mathcal{E}_L, \mathcal{E}_H, \mathcal{D}_Q \leftarrow$ LQ Encoder / HQ Encoder/ Decoder;
$\varphi \leftarrow$ Kalman Gain network;
$\Phi_{t-1\to t} \leftarrow$ Optical flow from previous frame;
$\omega \leftarrow$ Spatial warping operation;
$T \leftarrow$ length of clips;
Initialize $\tilde{\boldsymbol{z}}_1$, $\hat{\boldsymbol{z}}_1^+$, $\hat{\boldsymbol{y}}_1$;
for $t = 2, 3, \cdots, T$ **do**
  State Prediction:
  $\hat{\boldsymbol{z}}_t^- \leftarrow \mathcal{E}_H(\omega(\hat{\boldsymbol{y}}_{t-1}, \Phi_{t-1\to t}))$;
  State Update:
  $\tilde{\boldsymbol{z}}_t \leftarrow \mathcal{E}_L(\boldsymbol{x}_t)$;
  $\mathcal{K}_t \leftarrow \varphi(\tilde{\boldsymbol{z}}_1, \tilde{\boldsymbol{z}}_{t-1}, \tilde{\boldsymbol{z}}_t)$;
  $\hat{\boldsymbol{z}}_t^+ \leftarrow (1 - \mathcal{K}_t)\hat{\boldsymbol{z}}_t^- + \mathcal{K}_t\tilde{\boldsymbol{z}}_t$;
  $\hat{\boldsymbol{y}}_t \leftarrow \mathcal{D}_Q(\hat{\boldsymbol{z}}_t^+))$
end

A.3 Training Scheme.

Codebook Pre-Training (Stage I). Following CodeFormer [17], we first pretrain a codebook within a quantized autoencoder. Unlike TAST [5], the learned codebook is still image-based and does not involve temporal information. Precisely, given a HQ frame $\boldsymbol{y}_t \in \mathbb{R}^{H\times W\times 3}$ in pixel space, an encoder in HQ domain $\mathcal{E}_H$ encodes it into a latent code $\mathcal{E}_H(\boldsymbol{y}_t)$. Each token of the continuous code will be mapped to quantized discrete code $\hat{\boldsymbol{z}}_t^q$ from the learnable codebook $\mathcal{C} = \{c_k \in \mathbb{R}^d\}_{k=0}^N$ via nearest-neighbor matching. The decoder $\mathcal{D}$ then reconstructs the high-quality image frame from latent code. Similar to [4, 17], to train the quantized autoencoder, we adopt three image-level reconstruction losses: pixel loss$\mathcal{L}_1$, perceptual loss [6, 16] $\mathcal{L}_{per}$, and adversarial loss [11] $\mathcal{L}_{adv}$. Moreover, since image-level losses are underconstrained when updating the discrete codebook, code-level losses are also used to reduce the distance between the quantized code and input feature embeddings. The overall objectives in this stage are defined by

$$\mathcal{L}_I = \mathcal{L}_1 + \mathcal{L}_{per} + \mathcal{L}_{adv} + ||sg(\mathcal{E}_H(\boldsymbol{y}_t)) - \hat{\boldsymbol{z}}_t^q||_2^2 + ||\mathcal{E}_H(\boldsymbol{y}_t) - sg(\hat{\boldsymbol{z}}_t^q)||_2^2, \quad (3)$$

where $sg(\cdot)$ denotes stop-gradient operator.

Kalman Filter Network(Stage II). In this stage, we train a LQ encoder $\mathcal{E}_L$, the quantization Transformer T_q, and the Kalman filter network, while the codebook $\mathcal{C}$ and decoder $\mathcal{D}$ are frozen to preserve high-quality restoration from the VQGAN. Similar to [17], we adopt cross-entropy loss $\mathcal{L}_{CE}$ to supervise token prediction, and feature loss $\mathcal{L}_2$ to minimize the distance between features before and after quantization. The overall objectives are defined by

$$\mathcal{L}_{II} = \mathcal{L}_{CE} + \mathcal{L}_2(\mathcal{E}_H(\boldsymbol{y}_t), sg(\hat{\boldsymbol{z}}_t^q))). \quad (4)$$

Cross-Frame Attention (Stage III). To train both Cross-Frame Attention (CFA) modules and Controllable Feature Transformation (CFT), we fix other modules and use image-level reconstruction loss $\mathcal{L}_1$, $\mathcal{L}_{per}$ and GAN loss $\mathcal{L}_{adv}$, given by

$$\mathcal{L}_D = E[\log D(Y)] + E[1 - \log D(\hat{Y})]. \tag{5}$$

The discriminator D is constructed with multiple 3D convolution layers [3], denoted as temporal PatchGAN, which could further enhance the coherence of the generated face videos. The adversarial loss for the decoder modules is formulated as

$$\mathcal{L}_{adv} = -E[\log D(\hat{Y})]. \tag{6}$$

Additionally, we adopt temporal loss [7] between consecutive output frames, formulated as

$$\mathcal{L}_{temp} = \sum_{t=2}^{T} M_{t-1\to t} \cdot ||\hat{\boldsymbol{y}}_t - \hat{\boldsymbol{y}}_{t-1\to t}||_1, \tag{7}$$

where $M_{t-1\to t}$ denotes the valid mask computed by forward-backward consistency assumption [8], and $\hat{\boldsymbol{y}}_{t-1\to t}$ is the frame warped from previous frame $\hat{\boldsymbol{y}}_{t-1}$ with optical flow estimated by GT frames $\boldsymbol{y}_{t-1}$ and $\boldsymbol{y}_t$.

Hence, the overall training objectives are given by

$$\mathcal{L}_{III} = \lambda_1 \mathcal{L}_1 + \lambda_{per} \mathcal{L}_{per} + \lambda_{adv} \mathcal{L}_{adv} + \lambda_{temp} \mathcal{L}_{temp}. \tag{8}$$

Here λ_1, λ_{per}, λ_{adv}, and λ_{temp} are the balancing weights and we empirically set $\lambda_1 = 0.01$, $\lambda_{per} = 1$, $\lambda_{adv} = 0.1$, and $\lambda_{temp} = 0.1$.

A.4 Details of Dataset

Different from image-based degradations, video compression implicitly considers the dependencies across video frames, hence inducing temporal-variant degradations. This is implemented by randomly selecting codecs and constant rate factor (CRF) during training. The overall degradation model is defined by

$$\boldsymbol{x} = \{[(\boldsymbol{y} \circledast \boldsymbol{k}_\sigma) \downarrow + \boldsymbol{n}_\delta]_{codec}\} \uparrow, \tag{9}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are degraded and high-quality video clips, respectively. $\boldsymbol{k}$ and $\boldsymbol{n}_\delta$ are Gaussian blur kernel and Gaussian noise specified by σ and δ, respectively. $\circledast$ denotes the convolution operation, and $\downarrow$ and $\uparrow$ represent $4\times$ downsample and upsample in this paper. Video compression *codec* is selected from "libx264" and "h264" and the video quality is controlled by CRF, ranging from $[25, 45]$. During training, σ is sampled from $[2, 10]$, and noise level δ from $[0, 10]$.

For a comprehensive evaluation, we synthesize three splits of the VFHQ-Test dataset containing different levels of degradation. As summarized in Table A1, they follow the same degradation model but differ in the degree of noise, blur, and compression. Note that we mainly focus on the video compression controlled by CRF, which is unique for video tasks.

Besides synthetic degradations, we also assess the generalizability of our methods on real-world face videos. In particular, we collect 40 videos in the wild from YouTube, covering various degradations and celebrities in different scenes, *e.g.*, interviews, and talk shows. Given the raw video from online sources, data processing pipeline proposed by [14] is adopted to extract low-quality real face videos. For each video clip, we retain a sequence of 100 to 300 frames without scene transitions, which may break the dynamics between frames and hence deteriorate the temporal propagation.

Table A1: We divide the test data into different levels of difficulty for a more comprehensive analysis.

Degradation	mild	medium	heavy
Noise δ	[0, 5]	[5, 10]	[5, 10]
Blur σ	[2, 5]	[5, 10]	[5, 10]
CRF	[18, 25]	[25, 35]	[35, 45]

B More Analysis

B.1 Effectiveness of Alignment

Alignment is the common pre-processing procedure in face-related vision tasks. This ensures the faces are transformed and centralized in the same canonical coordinate system. This is realized by detecting landmark keypoints and applying affine transformation to the original face images, which is sensitive to the locations of detected facial landmarks. Mild inaccuracy of landmark detections is tolerable in a single image. However, the noisy detections could consequently result in unintentional temporal inconsistencies between frames in a video.

To reduce the additional inconsistency, we adopt low-pass Gaussian filter on the locations of each landmarks along the temporal dimension, which eliminates abrupt change (jitters) oriented along time. Denoted $\mathcal{M}_t^k$ as the k-th detected landmarks from frame y_t, where t represents the timestep. The filtered landmarks are given by

$$\hat{\mathcal{M}}_t^k = \sum_{n=t-r}^{t+r} G(n, \sigma) \cdot \mathcal{M}_n^k, \tag{10}$$

where $G(n, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(n-t)^2}{2\sigma^2}}$, and r denotes the radius of the window size. In our experiments, we empirically set $\sigma = 5$ and $r = 20$. Figure A3 provides a visual example of landmarks processed by Gaussian filter. The temporal jitters are largely alleviated by the filter. We also demonstrate the effectiveness of alignment in the supplementary video.

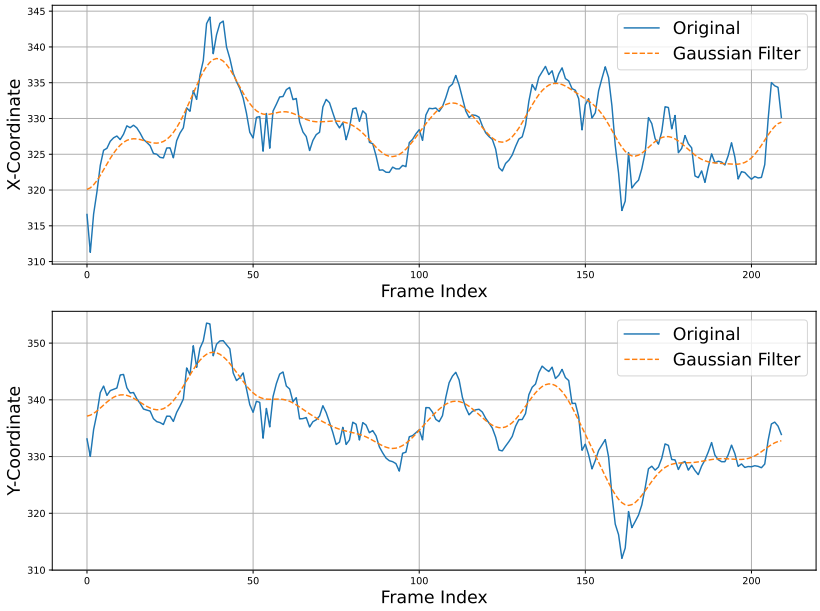

Fig. A3: A representative example of landmark location processed by Gaussian filter along time.

B.2 Quantitative Comparison on Various Degradation.

Table A2 provides the full quantitative results of models on different test partitions. As can be observed, our proposed method consistently outperforms all other concurrent approaches on all datasets. In particular, KEEP surpasses BasicVSR++ [2] by a large margin of 0.95 dB in PSNR on test dataset with heavy degradation. For identity preservation and pose quality metrics (*IDS* and *AKD*), our method achieves top performance and fewer fluctuations. On VFHQ-mild dataset, KEEP possesses 8.82 average keypoint distances on images of shape 512×512, while the distances of all other methods are over 10.53. This suggests that our method could better preserve identity within the generated video and introduce far less jitters in the pose of faces. Such improvements are significant in VFSR.

C Limitations and Future Work

Figure A4 presents a failure case of our method when the input video suffers heavy degradation. The recovered logo on the hat in different frames shows inconsistent shapes. This could stem from the inherent limitation that the contents in non-facial areas are unstructured and highly deviate from what the facial prior code encapsulates. A potential solution is to use general well-trained VSR mod-

Table A2: Quantitative comparison on VFHQ dataset with different levels of degradation. Red and Blue indicate the best and the second best results.

Method	PSNR↑	SSIM↑	LPIPS↓	IDS↑	AKD↓	$\sigma_{IDS}(\times 10^{-2})\downarrow$	$\sigma_{AKD}\downarrow$
				Mild			
GPEN [15]	25.5193	0.7517	0.2988	0.7142	11.4691	4.7416	3.5109
GFPGAN [10]	26.2933	0.7795	0.2482	0.7437	10.5467	4.5700	3.6482
RestoreFormer [12]	25.5720	0.7344	0.3195	0.7530	10.5354	4.7159	3.4122
CodeFormer [17]	24.6597	0.7454	0.2742	0.6272	11.4983	6.3726	3.6927
EDVR [9]	26.6051	0.7858	0.2484	0.7195	11.6220	4.8048	3.5829
BasicVSR [1]	26.0458	0.7765	0.2496	0.6973	11.3679	5.0343	3.6054
BasicVSR++ [2]	27.1996	0.8057	0.1958	0.7641	11.3136	5.2543	4.6425
KEEP (Ours)	27.9994	0.8267	0.1619	0.7960	8.8182	3.6866	3.2538
				Medium			
GPEN [15]	25.1871	0.7460	0.3063	0.6741	12.2091	5.3168	3.6872
GFPGAN [10]	26.2826	0.7839	0.2554	0.6970	11.1332	5.2539	3.7173
RestoreFormer [12]	25.5123	0.7256	0.3346	0.7044	11.2567	5.3760	3.6448
CodeFormer [17]	24.6238	0.7424	0.2852	0.6077	11.8149	6.7256	3.7899
EDVR [9]	26.3385	0.7815	0.2625	0.6771	12.4233	5.2598	3.6660
BasicVSR [1]	25.8332	0.7725	0.2594	0.6638	12.4503	5.7990	3.8101
BasicVSR++ [2]	26.5465	0.7918	0.2203	0.6919	13.4386	6.8957	5.6914
KEEP (Ours)	27.4853	0.8171	0.1740	0.7481	9.5937	4.6179	3.3764
				Heavy			
GPEN [15]	25.0191	0.7437	0.3108	0.6544	12.4814	5.6768	3.8088
GFPGAN [10]	26.0747	0.7807	0.2613	0.6761	11.6804	6.8689	3.9346
RestoreFormer [12]	25.3354	0.7216	0.3458	0.6715	11.7674	5.6277	3.6966
CodeFormer [17]	24.5600	0.7407	0.2916	0.5949	12.0462	5.9110	3.9079
EDVR [9]	26.1600	0.7792	0.2729	0.6524	13.0927	5.9243	3.8166
BasicVSR [1]	25.6895	0.7695	0.2686	0.6426	12.7841	6.1689	3.8356
BasicVSR++ [2]	26.2686	0.7872	0.2289	0.6650	14.2254	7.2980	6.1919
KEEP (Ours)	27.2165	0.8124	0.1803	0.7282	9.8833	4.8643	3.3217

els to further enhance these regions and backgrounds. We leave this avenue of research as future work.

D Evaluation on Real-World Videos

Figure A5 shows that our method recovers texture details in each frame. In addition, the supplementary video delivers high-quality restoration of our method with superior consistency from highly degraded face videos, demonstrating extraordinary generalization in face videos in the wild.

Fig. A4: Limitations. Our method might produce inconsistent results on non-facial areas when the input video exhibits heavy degradation. For example, the logo on the hat shows various shapes in different frames.

E Additional Visual Results

Figure A6, A7, A8, and A9 showcase additional visual examples of our methods and other compared baselines.

References

1. Chan, K.C., Wang, X., Yu, K., Dong, C., Loy, C.C.: BasicVSR: The search for essential components in video super-resolution and beyond. In: CVPR (2021)
2. Chan, K.C., Zhou, S., Xu, X., Loy, C.C.: Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. In: CVPR (2022)
3. Chang, Y.L., Liu, Z.Y., Lee, K.Y., Hsu, W.: Free-form video inpainting with 3d gated convolution and temporal patchgan. In: ICCV (2019)
4. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: CVPR (2021)
5. Ge, S., Hayes, T., Yang, H., Yin, X., Pang, G., Jacobs, D., Huang, J.B., Parikh, D.: Long video generation with time-agnostic vqgan and time-sensitive transformer. In: European Conference on Computer Vision. pp. 102–118. Springer (2022)
6. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: ECCV (2016)
7. Lai, W.S., Huang, J.B., Wang, O., Shechtman, E., Yumer, E., Yang, M.H.: Learning blind video temporal consistency. In: ECCV (2018)
8. Sundaram, N., Brox, T., Keutzer, K.: Dense point trajectories by gpu-accelerated large displacement optical flow. In: ECCV (2010)

9. Wang, X., Chan, K.C., Yu, K., Dong, C., Change Loy, C.: EDVR: Video restoration with enhanced deformable convolutional networks. In: CVPRW (2019)
10. Wang, X., Li, Y., Zhang, H., Shan, Y.: Towards real-world blind face restoration with generative facial prior. In: CVPR (2021)
11. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., Change Loy, C.: ESRGAN: Enhanced super-resolution generative adversarial networks. In: ECCVW (2018)
12. Wang, Z., Zhang, J., Chen, R., Wang, W., Luo, P.: Restoreformer: High-quality blind face restoration from undegraded key-value pairs. In: CVPR (2022)
13. Wu, J.Z., Ge, Y., Wang, X., Lei, S.W., Gu, Y., Shi, Y., Hsu, W., Shan, Y., Qie, X., Shou, M.Z.: Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In: ICCV (2023)
14. Xie, L., Wang, X., Zhang, H., Dong, C., Shan, Y.: VFHQ: A high-quality dataset and benchmark for video face super-resolution. In: CVPR (2022)
15. Yang, T., Ren, P., Xie, X., Zhang, L.: Gan prior embedded network for blind face restoration in the wild. In: CVPR (2021)
16. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
17. Zhou, S., Chan, K.C., Li, C., Loy, C.C.: Towards robust blind face restoration with codebook lookup transformer. In: NeurIPS (2022)

Fig. A5: Qualitative comparison on the real face videos. Our KEEP recovers high-fidelity face videos with faithful and consistent details.

Fig. A6: Qualitative comparison on the VFHQ. RFormer represents RestoreFormer [12].

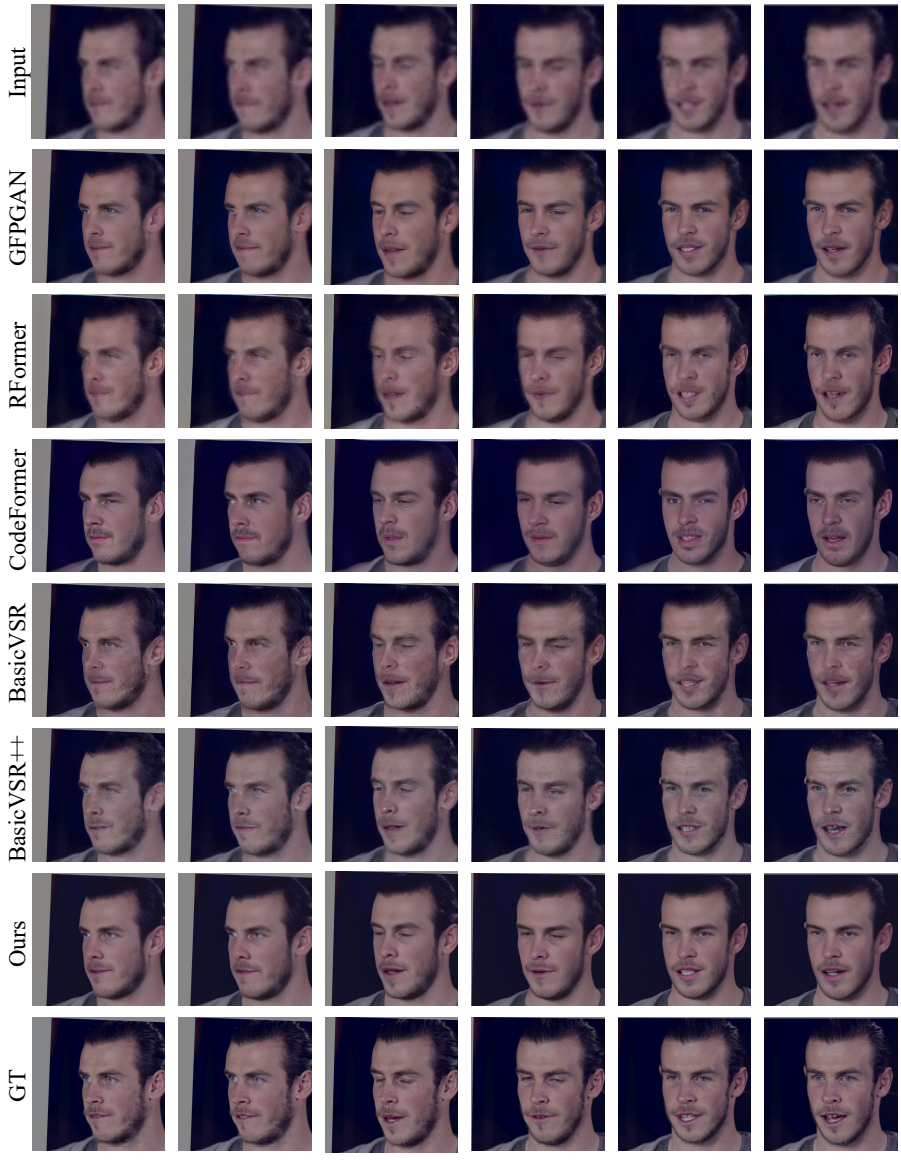

Fig. A7: Qualitative comparison on the VFHQ. RFormer represents RestoreFormer [12].

Fig. A8: Qualitative comparison on the VFHQ. RFormer represents RestoreFormer [12].

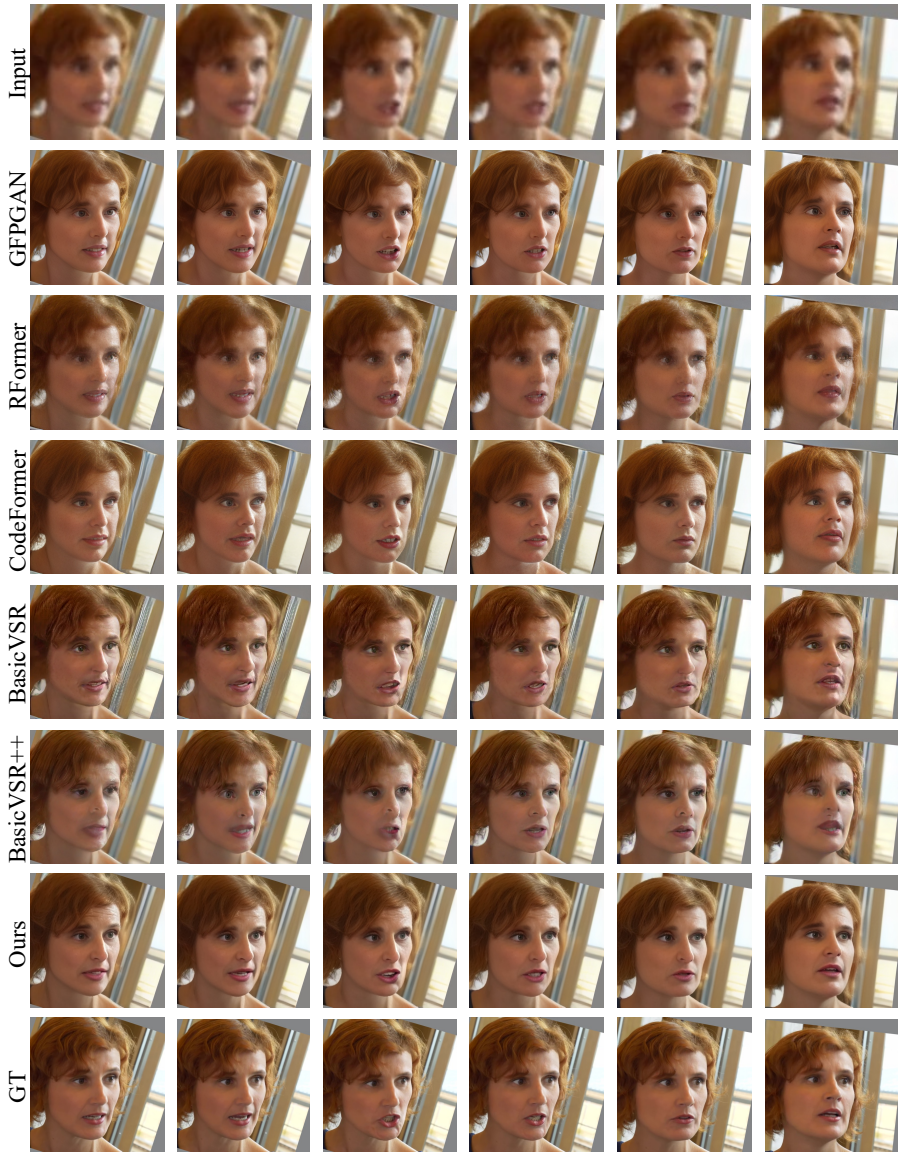

Fig. A9: Qualitative comparison on the VFHQ. RFormer represents RestoreFormer [12].