Supplements for FLAIR: A Conditional Diffusion Framework with Applications to Face Video Restoration

Zihao Zou[1,*], Jiaming Liu[2,*], Shirin Shoushtari[2], Yubo Wang[2], and Ulugbek S. Kamilov[2]
[1]University of North Carolina, Chapel Hill, NC, USA
[2]Washington University in St. Louis, MO, USA
[*]These authors contributed equally.

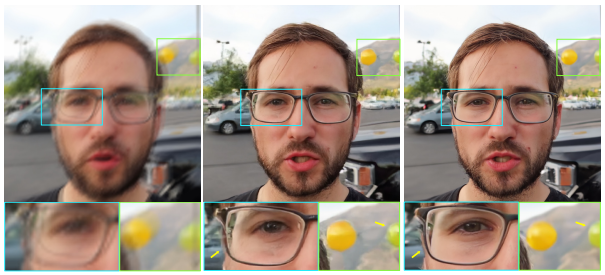

Figure 1. Visual illustration of the impact of equation (3) on training image DPMs. The zoomed-in regions are shown below the main results. Notably, the image restoration quality is improved by applying data augmentation to the conditional inputs.

1. Additional Implementation Details

In this section, we present additional implementation details omitted from the main paper due to space constraints. We train and evaluate all models with Pytorch on a computing cluster equipped with A40-40GB and A100-80GB GPUs. Our implementation code and pre-trained model (download link) can be found in the supplements. The parameter settings are presented in Table 7. The size of our video DPM is 1310 MB on the disk, while that of image DPM is 576 MB.

1.1. Training of conditional Image DPMs

In order to improve the generation flexibility and empirical performance of FLAIR, we jointly train a single image diffusion model on conditional and unconditional objectives by randomly dropping $\boldsymbol{c}$ during training (*e.g.*, $p_{\text{uncond}} = 0.2$), similar to the *classifier free guidance* [12, 17]. Hence, the sampling is performed using the adjusted noise prediction:

$$\tilde{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{c}, t) = \lambda \epsilon_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{c}, t) + (1-\lambda)\epsilon_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t), \quad (1)$$

where $\lambda > 0$ is the trade-off parameter, and $\epsilon_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ is the unconditional $\boldsymbol{\epsilon}$-prediction. For example, setting $\lambda = 1$ disables the unconditional guidance, while increasing $\lambda > 1$ strengthens the effect of conditional $\boldsymbol{\epsilon}$-prediction. The objective function for training the $\tilde{\epsilon}_{\boldsymbol{\theta}}$ is

$$\mathcal{L}_{\boldsymbol{\theta}} = \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{c}, \boldsymbol{\epsilon}, t\sim[1,T]} \left[\|\boldsymbol{\epsilon} - \tilde{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{c}, t)\|^2 \right]. \quad (2)$$

Given that our video diffusion restoration models are fine-tuned on pre-trained image DPMs, it is reasonable to assume that a superior pre-trained image DPM would result in an better video DPM in terms of restoration quality. To this end, a data augmentation for training conditional image DPMs is done by constructing the conditional inputs $\boldsymbol{c} \in \mathbb{R}^{Nd}$ as follows

$$\boldsymbol{c} = \boldsymbol{m}_c \odot (\boldsymbol{y}) \uparrow^s_{bicubic}, \quad (3)$$

where $\boldsymbol{m}_c$ is a weighted mask that randomly reduces the importance of some pixels, analog to the masked augmentation training proposed in [11]. We have observed that this data augmentation on $\boldsymbol{c}$ can improve the restoration results especially on large motion degradation, as shown in Fig. 1. The conditional input $\boldsymbol{c}$ is normalized to intensity range of $[-1, 1]$ for better performance and stable training. We train all image DPMs in half precision (`float16`) with a batch-size of 64. We use the Adam optimizer with a fixed learning rate of 1.5×10^{-4} and a dropout rate of 0.2 for each model. In Fig. 5, we present samples of synthetically generated random kernels, following [1, 26], used to generate the image and video deblurring dataset.

1.2. Implementations of Video DPM

We use `einops` [16] to efficiently rearrange the features between spatial and temporal layers.

Group Normalization for Sequential Features. For video DPMs, we observe that directly calculating group normalization to video features as independent images by rearranging the input as $\mathbb{R}^{B\times N\times C\times H\times W} \rightarrow \mathbb{R}^{(BN)\times C\times H\times W}$ results in temperature unalignment across frames. When calculating the group normalization, we consider the entire video by rearranging the input from $\mathbb{R}^{B\times N\times C\times H\times W}$ to $\mathbb{R}^{B\times C\times N\times H\times W}$, Consequently, the group normalization is computed along the N, H, W axis. We have observed that

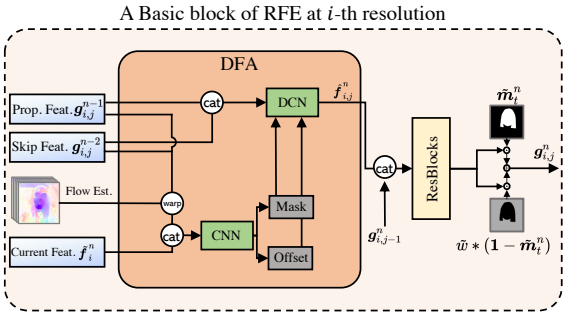

Figure 2. Illustration of one basic block of our proposed recurrent feature enhancement (RFE) module. The `cat` operator denotes feature concatenation.

applying this rearrangement to group normalization layers, which are pre-trained in image DPM, does not result in any performance degradation.

More details about RFE Module. As introduced in the main paper, we implement recurrent feature enhancement (RFE) module to capture sequential dependencies and synchronize video frame features at high resolutions (*e.g.,* [512, 256]). Fig 2 illustrates one basic block of our RFE module. Given the extracted temporal features $\{\tilde{\boldsymbol{f}}_i^n\}_{n=1}^N$ from the 3D residual blocks at i-th resolution scale, we apply Deformable Feature Alignment (DFA) [4] to propagate and align the intermediate features $\hat{\boldsymbol{f}}_{i,j}^n$ as

$$\hat{\boldsymbol{f}}_{i,j}^n = \texttt{DFA}(\tilde{\boldsymbol{f}}_i^n, \boldsymbol{g}_{i,j}^{n-1}, \boldsymbol{g}_{i,j}^{n-2}, \boldsymbol{o}_i^{n\to n-1}, \boldsymbol{o}_i^{n\to n-2}),$$

where $\boldsymbol{g}_{i,j}^{n-1}$ and $\boldsymbol{g}_{i,j}^{n-2}$ are the features at the $(n-1)$-th and $(n-2)$-th sequential step in the j-th propagation branch, respectively. For example, we have $\boldsymbol{g}_{i,0}^n = \tilde{\boldsymbol{f}}_i^n$. Similarly, the $\boldsymbol{o}_i^{n_1\to n_2}$ denotes the optical flow estimated from n_1-th degraded input frame to the n_2-th counterparts. The features $\hat{\boldsymbol{f}}_{i,j}^n$ are then concatenated (`cat`) and passed into a stack of residual blocks (`ResBlocks`) to fuse $\boldsymbol{g}_{i,j}^n$, denoted as

$$\tilde{\boldsymbol{g}}_{i,j}^n = \hat{\boldsymbol{f}}_{i,j}^n + \texttt{ResBlocks}(\texttt{cat}(\boldsymbol{g}_{i,j-1}^n, \hat{\boldsymbol{f}}_{i,j}^n)), \quad (4)$$

$$\boldsymbol{g}_{i,j}^n = \tilde{w} * (\mathbf{1} - \tilde{\boldsymbol{m}}_t^n) \odot \tilde{\boldsymbol{g}}_{i,j}^n + (\tilde{\boldsymbol{m}}_t^n) \odot \tilde{\boldsymbol{g}}_{i,j}^n, \quad (5)$$

where $\tilde{w} \in [0, 1]$ balances the smoothness of the background scenes of the fused featur, denoted as $(\mathbf{1} - \tilde{\boldsymbol{m}}_t^n) \odot \tilde{\boldsymbol{g}}_{i,j}^n$. The masks $\{\tilde{\boldsymbol{m}}_t^n\}_{n=1}^N$ are the downscale version of facial region masks $\boldsymbol{m}_t = \{\boldsymbol{m}_t^n\}_{n=1}^N$ estimated from $\boldsymbol{x}_{0t}$ at the t-th reverse diffusion step. The main motivation behind the design of propagation annealing is to enhance robustness against appearance changes and error accumulation within the recurrent network. We have observed that this annealing can notably improve the temporal consistency of background scenes across frames while preserving the sharpness of facial region, as shown in Fig 3.

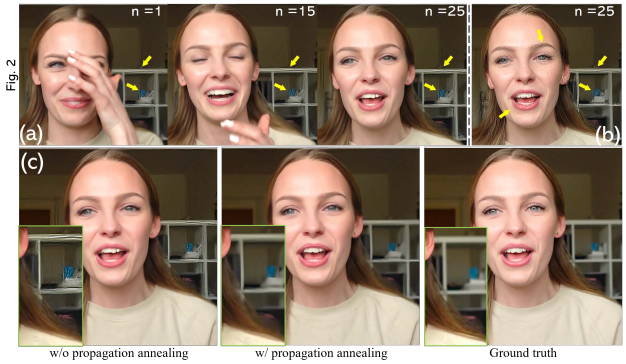

Figure 3. Visual demonstration of the impact of our propagation annealing in (4) on $8\times$ SR task. Our RFE layers reuse the same recurrent blocks to save parameters and reduce memory by predicting new frame features from previously refined ones. As shown in [13], this sequential strategy leads to noise accumulation in long-range data. In *(a)*, static background artifacts accumulate more easily than moving face without annealing. In *(b)*, repeating 1 frame 25 times shows artifacts in the static face part without annealing. The same annealing setting is used for all experiments in the paper, indicating this strategy's generalizability. In *(c)*, the background scenes in each frame are improved due to the use of our propagation annealing, as shown in the zoomed-in figure.

1.3. Training of video DPMs

All video DPMs are fine tuned with batch size $B = 4$ and frame length $N = 10$. We set schedule $T = 1000$ and uniformly spaced β_t for both video deblurring and JPEG restoration, while $T = 2000$ for video super-resolution tasks. We use the Adam optimizer with a fixed learning rate of 1×10^{-4} and weight-decay of 0.05 for fine-tuning the video DPMs. Similarly, we train all DPMs in half precision (`float16`). We do not apply classifier free guidance for fine-tuning video diffusion model. Note that, we do not perform any checkpoint selection on our models and simply select the latest checkpoint of each model. It will take around a week to get a video DPM.

1.4. Implementations during Inference

Our proposed reverse diffusion sampling is illustrated in Algorithm 1. We use an exponential decay for γ_t, where we parameterize $\gamma_t = 1 - \zeta\frac{\sigma_e^2\bar{\alpha}_t}{\bar{\alpha}_{t-1}}$, where ζ controls the strength of the data consistency module, and γ is clipped into range $[0, 1]$. The setting of ζ for each task is presented in Table 7. We use an exponential growth for $\{w_t\}_{t=\tau}^{K-1}$. We parameterize $w_t = e^{-(t-\tau)/(K-\tau)} * w_\tau$, where w_τ controls the final strength of the enhancement module, and τ controls where the enhancement modules end its participation during sampling. The setting of w_τ and τ for each task can be found in Table 7. We run a grid search for best controlling hyperparameters of the two-stage conditional refinement and the rescheduling time step K for each dataset, similar

Algorithm 1 FLAIR Face Video Iterative Refinement

Input: $\epsilon_{\boldsymbol{\theta},\boldsymbol{\phi}}$: Video denoiser network; $\boldsymbol{y}$: Degraded video; $\mathcal{G}$: Image Enhancement module; γ_t, ρ_t, w_t;
Output: Restored video $\boldsymbol{x}_0$
Sample $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ Run diffusion sampling
for $t = T, \ldots, 1$ **do**
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\boldsymbol{x}_{0t} = \frac{1}{\sqrt{\bar{\alpha}_t}}(\boldsymbol{x}_t + (1-\bar{\alpha}_t)\epsilon_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x}_t, \boldsymbol{c}, t))$
  $\tilde{\boldsymbol{x}}_{0t} = \boldsymbol{x}_{0t} - \gamma_t(\mathcal{A}^{+}\mathcal{A}\boldsymbol{x}_{0t} - \mathcal{A}^{+}\boldsymbol{y})$
  $\tilde{\boldsymbol{x}}_{0t} = (\mathbf{1} - w_t\boldsymbol{m}_t) \odot \tilde{\boldsymbol{x}}_{0t} + w_t\boldsymbol{m}_t \odot \mathcal{G}(\tilde{\boldsymbol{x}}_{0t})$
  $\tilde{\boldsymbol{\epsilon}}_t = \frac{1}{\sqrt{1-\bar{\alpha}_t}}(\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\tilde{\boldsymbol{x}}_{0t})$
  $\boldsymbol{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\tilde{\boldsymbol{x}}_{0t} + \sqrt{1-\bar{\alpha}_t}(\sqrt{1-\rho_t}\tilde{\boldsymbol{\epsilon}}_t + \sqrt{\rho_t}\boldsymbol{\epsilon})$
end for
return: $\boldsymbol{x}_0$

to [20,22,23,29]. This inference-time hyperparameter tuning is cheap as it does not involve retraining or fine-tuning the model itself. The facial mask $\boldsymbol{m}_t$ estimation follows the similar method as [10, 22, 24, 27], where we introduce in a separate subsection 1.6.

1.5. Baseline Methods

CodeFormer [27], VQFR [10] and RestoreFormer++ [21] refer to recently developed conditioning generative methods that use pre-trained Vector-Quantization (VQ) codebooks as dictionaries, achieving SOTA results in blind face restoration. These codebooks are learned on the entire facial region. We employ their original implementations [1,2,3] and pre-trained models for our tasks. For all these three baseline methods, we follow their original implementations of frame background enhancement accordingly.

VRT [13] denotes a recently developed video restoration transformer (VRT) method, characterized by its parallel frame prediction and long-range temporal dependency modeling abilities. VRT has been shown superior performance for general restoration tasks such as video denoising, deblurring, super-resolution, etc. We modify the publicly available implementation [4] and train the model for each task on the same CelebV-Text [25] video training dataset as FLAIR.

BasicVSPP [4] is another recent SOTA method based on recurrent refinement structure for video super-resolution. BasicVSPP improves over BasicVSR [3] by proposing a second-order grid propagation with flow guided deformable alignment. Likewise, we modify the publicly available implementation [5] and train the model on the same CelebV-Text [25] training dataset as FLAIR.

ILVR [7] and DR2E [22] are two recently developed conditioning methods based on unconditionally trained image DPM for solving versatile blind image restoration tasks. Both ILVR and DR2E share the similar conditional sampling implementation, whereas DR2E adapts an additional enhancement module for face regions similar to FLAIR. We modify the publicly available implementation [6,7] of both methods for each FVR task. We use the similar grid search to FLAIR for fine-tuning the hyper-parameters within ILVR and DR2E, respectively.

DDNM [7] and DiffPIR [22] refer to recently developed conditioning methods based on unconditionally trained image DPM for solving general image inverse problems. Unlike ILVR and DR2E, DDNM and DiffPIR rely on the forward-model to impose data-consistency. Similarly, we modify the publicly available implementation [8,9] of both methods for each FVR task. We use the similar grid search to FLAIR for fine-tuning the hyper-parameters within DDNM and DiffPIR, respectively.

We pre-train an unconditional image DPM on FFHQ and then fine tune it on the same CelebV-Text images used for video DPMs as additional baseline. All diffusion model based baseline methods, including ILVR, DR2E, DDNM, DiffPIR share the same unconditional image DPM. We train the baseline unconditional diffusion model modified based on the publicly available PyTorch implementation [10] for around 1×10^7 samples in total (pre-training and fine-tuning).

1.6. Face Detection and Processing

We process the images using the tools provided in `facexlib`[11].

Face Region Affine Transformation. We first use `RetinaFace` [9] to calculate the face landmarks. Then we use `OpenCV` [2] to estimate affine matrices and transform the images to the head-only version with bicubic interpolation.

Estimation of Face Mask $\boldsymbol{m}_t$. We use `ParseNet` [5] to get the face parsing map, and convert it to a soft mask $\boldsymbol{m}_t$ with Gaussian blurring. The above process has been widely adapted for FVR in recent methods, such as [10, 19, 21, 22, 24, 27].

2. Datasets

CelebV-HQ [28] dataset is a large-scale, high-quality video dataset with rich facial attributes for video generation and editing. CelebV-HQ contains 35,666 video clips with the

[1] https://github.com/sczhou/CodeFormer
[2] https://github.com/TencentARC/VQFR
[3] https://github.com/wzhouxiff/RestoreFormerPlusPlus
[4] https://github.com/JingyunLiang/VRT
[5] https://github.com/open-mmlab/mmagic
[6] https://github.com/jychoi118/ilvr_adm
[7] https://github.com/Kaldwin0106/DR2_Drgradation_Remover
[8] https://github.com/wyhuai/DDNM
[9] https://github.com/yuanzhi-zhu/DiffPIR
[10] https://github.com/openai/guided-diffusion
[11] https://github.com/xinntao/facexlib

Method	Task	CelebV-Text [25]						CelebV-HQ [28]					
		PSNR	SSIM	LPIPS	FVD	FID	KID	PSNR	SSIM	LPIPS	FVD	FID	KID
VQFR [10]	8× Bicubic	26.34	0.805	0.221	238.89	46.53	9.92	26.37	0.793	0.219	528.02	74.01	14.76
CodeFormer [27]		26.60	0.783	0.238	215.07	50.03	12.40	26.64	0.770	0.236	444.52	81.58	20.44
RestoreFormer++ [21]		27.13	0.792	0.225	130.64	42.64	8.58	27.69	0.790	0.208	330.02	61.94	14.26
DR2E [22]		26.59	0.810	0.220	243.15	46.62	10.95	26.56	0.798	0.216	556.67	73.16	15.22
FLAIR (Ours)		**32.13**	**0.889**	**0.139**	**62.43**	**31.93**	**6.29**	**31.80**	**0.875**	**0.132**	**146.57**	**42.06**	**6.68**
VQFR [10]	16× Bicubic	24.31	0.762	0.270	383.47	55.04	13.69	24.28	0.743	0.268	797.95	88.40	19.94
CodeFormer [27]		24.39	0.732	0.298	397.34	59.57	16.20	24.37	0.713	0.302	865.36	98.22	25.64
RestoreFormer++ [21]		23.70	0.719	0.295	284.66	56.20	12.17	24.36	0.715	0.279	615.80	89.85	19.77
DR2E [22]		24.23	0.755	0.271	400.64	51.95	12.45	24.33	0.741	0.266	722.86	84.81	17.62
FLAIR (Ours)		**28.49**	**0.844**	**0.230**	**201.86**	**50.73**	**10.24**	**28.31**	**0.808**	**0.216**	**413.81**	**78.38**	**11.68**

Table 1. Quantitative results calculated only within face regions on two video datasets (short clips). VQFR, CodeFormer, RestoreFormer++ and DR2E are SOTA face restoration methods that rely on separate methods for backgrounds enhancement. Note the quantitative improvements achieved by FLAIR when it is specifically evaluated on face regions. **Best** and second-best values for each metric are color-coded.

Method	PSNR↑	SSIM↑	LPIPS↓	FVD↓	FID↓	KID↓
4×, Motion blur, $\sigma = 0.05$						
$\mathcal{A}^+ y$	14.62	0.244	0.850	3515.79	200.59	134.43
VRT [13]	30.58	**0.904**	0.173	149.73	68.94	26.95
CodeFormer [27]	27.74	0.817	0.188	596.37	65.90	19.70
RestoreFormer++ [21]	27.88	0.819	0.189	587.97	64.66	18.25
VQFR [10]	27.21	0.808	0.205	836.61	75.00	22.84
DR2E [22]	27.04	0.799	0.213	1135.91	76.98	22.72
DiffPIR [29]	29.55	0.855	0.213	1139.93	51.59	12.41
DDNM [20]	29.21	0.847	0.267	762.26	95.58	42.09
FLAIR (Ours)	31.10	0.890	0.151	**126.24**	48.21	15.36
FLAIR-SA (Ours)	**31.66**	0.897	0.152	131.54	49.68	17.97
FLAIR+CodeFormer (Ours)	31.12	0.891	0.147	127.43	47.17	14.89
FLAIR+RestoreFormer++ (Ours)	31.03	0.876	**0.146**	134.59	**43.66**	**12.25**

Table 2. Quantitative results of motion blur on CelebV-Text [25] (long clips). **Best** and second-best values for each metric are color-coded.

resolution of 512×512 at least. All data is publicly available [12]. We randomly select 20 clips, each containing 25 high quality sequences from CelebV-HQ.

CelebV-Text [25] dataset is another large-scale, high-quality, diverse dataset of facial text-video pairs. CelebV-Text comprises $70,000$ in-the-wild face video clips with diverse visual content. All data is publicly available [13]. we select 7200 clips with each containing 20 high quality 512×512 sequences for training. For video testing datasets, we randomly chose 125 short clips and 6 long clips from the unused portion of the CelebV-Text, ensuring no identity overlap with the fine-tuning datasets. Each short clip contains 25 sequences, and each long clip contains 100 sequences. As highlighted by its original authors, the videos that have appeared in CelebV-HQ are filtered out.

Web Video Clip. We extract four low quality web videos of around 150 frames from YouTube following [6], which suffers from complex unknown degradation. The collected clip is then crop out the face-only region using the same processes as in 1.6, following [10, 24, 27].

3. Additional Results

We present additional experimental results that were omitted from the main paper due to space limitations. *We provide several video comparisons of our FLAIR in the supplementary materials.*

3.1. Additional Numerical Results

Numerical Evaluation on Facial Region Only. Given that some of the state-of-the-art (SOTA) methods, including VQFR, CodeFormer, RestoreFormer++, and DR2E, are primarily designed for face restoration and utilize separate backbones for background enhancement, we have conducted additional numerical comparison for resorting facial region only. In Table 1, we report the PSNR, SSIM, LPIPS, FVD, FID, and KID results for $8\times$ and $16\times$ video super-resolution on the short clips of CelebV-Text and CelebV-HQ datasets, respectively. As expected, our FLAIR quantitatively outperforms all other baseline methods in terms of both perception and data-fidelity metrics.

Other Quantitative Results. In Table 3, we report numerical results of FLAIR and some baseline methods for $4\times$ video super-resolution on two datasets. Note the better performance achieved by our FLAIR with different enhancement backbones even under mild degradation. To further show that there is potential to adapt versatile backbones for our FLAIR enhancement module, we report numerical results of our FLAIR using the same pre-trained unconditional image DPM in (1) as our enhancement backbone for $4\times$ SR, noisy Gaussian deblurring task. To demonstrate the adaptability of various backbones for our FLAIR enhancement module, we present numerical results where FLAIR employs the same pre-trained unconditional image DPM, as referenced in (1), as its enhancement backbone. For simplicity, we have limited our experiments to $4\times$ SR, noisy Gaussian deblurring task, deferring a more comprehensive evaluation to future work. The visual comparisons are shown in Fig 4. We make an interesting observation that FLAIR using unconditional image DPM as face enhancement module can improve the final restoration results in terms of PSNR and FVD on CelebV-Text.

Evaluation of Running Time. For completeness, we also report the running time of our FLAIR compared with the other

[12] https://celebv-hq.github.io/
[13] https://celebv-text.github.io/

Method	CelebV-Text [25]						CelebV-HQ [28]					
	PSNR	SSIM	LPIPS	FVD	FID	KID	PSNR	SSIM	LPIPS	FVD	FID	KID
VQFR [10]	28.88	0.855	0.160	151.86	46.25	10.34	28.59	0.847	0.156	261.27	66.98	14.50
CodeFormer [27]	29.80	0.867	0.153	107.39	45.46	10.6	29.17	0.856	0.151	219.77	66.42	15.53
RestoreFormer++ [21]	29.06	0.856	0.151	111.53	45.80	10.21	28.96	0.849	0.149	211.02	65.51	12.60
DR2E [22]	28.40	0.836	0.167	189.91	44.49	9.18	27.98	0.800	0.163	378.15	76.39	15.33
DDNM [20]	34.76	0.929	0.118	31.48	37.65	20.28	33.46	0.917	0.129	89.33	55.27	27.89
FLAIR (Ours)	**36.05**	**0.942**	0.061	26.57	11.27	2.64	**34.46**	**0.932**	0.060	76.18	15.36	1.50
FLAIR+CodeFormer (Ours)	35.10	0.934	0.059	**26.44**	**9.51**	**0.75**	33.47	0.920	0.059	**74.56**	**13.84**	**0.04**
FLAIR+RestoreFormer++ (Ours)	35.42	0.936	**0.057**	27.22	10.24	1.59	34.17	0.927	**0.056**	78.07	14.49	0.69

Table 3. Quantitative results of $4\times$ face video super-resolution on two separate video datasets (short clips). Note the quantitative improvements achieved by integrating our enhancement module within FLAIR, even in cases of mild degradation. **Best** and second-best values for each metric are color-coded.

Method	Task	CelebV-Text [25]						CelebV-HQ [28]					
		PSNR↑	SSIM↑	LPIPS↓	FVD↓	FID↓	KID↓	PSNR↑	SSIM↑	LPIPS↓	FVD↓	FID↓	KID↓
$\mathcal{A}^+\boldsymbol{y}$		21.40	0.740	0.412	481.22	202.14	218.43	22.25	0.731	0.424	863.97	256.04	257.19
VQFR [10]		26.40	0.801	0.255	229.86	76.46	23.24	25.81	0.777	0.277	482.91	126.86	41.15
RestoreFormer++ [21]		26.48	0.799	0.249	190.48	70.39	16.81	25.98	0.775	0.273	470.12	123.14	38.55
CodeFormer [27]		26.66	0.798	0.259	214.37	76.11	21.39	26.00	0.775	0.278	498.19	126.28	39.34
DR2E [22]	8× Bicubic	27.89	0.824	0.202	205.48	53.68	13.51	27.49	0.8073	0.207	419.64	91.28	22.86
DDNM [20]		29.95	0.860	0.234	122.03	72.16	44.07	29.00	0.836	0.253	352.08	113.65	64.10
ILVR [7]		29.62	0.852	0.206	145.22	52.72	21.39	28.77	0.829	0.222	350.38	90.95	37.88
FLAIR (Ours)		**30.76**	**0.868**	**0.159**	**75.16**	**41.46**	**8.11**	**29.56**	**0.844**	**0.157**	**194.79**	**66.69**	**13.79**

Table 4. Quantitative results on two face video datasets (short clips). Our method generates better perceptual quality and data-fidelity results than SOTA face restoration baselines. **Best** and second-best values for each metric are color-coded.

Method	PSNR	SSIM	LPIPS	FVD	FID	KID
CelebV-Text [25] (short clips)						
FLAIR (Ours)	29.87	0.856	0.149	82.82	39.54	8.25
FLAIR+Unconditional DPM (Ours)	30.73	0.865	0.157	81.09	45.48	12.65
CelebV-Text [25] (long clips)						
FLAIR (Ours)	31.51	0.858	0.169	175.52	55.88	20.85
FLAIR+Unconditional DPM (Ours)	31.44	0.859	0.163	146.31	55.69	20.95

Table 5. Quantitative results of FLAIR using unconditional image DPM as enhancement module for $4\times$ super-resolution, Gaussian deblurring, AWGN $\sigma = 0.05$ on CelebV-Text [25].

Method	Sampling Time (sec)
DDNM [20]	42.95
ILVR [7]	101.21
FLAIR (Ours)	112.53
FLAIR+CodeFormer (Ours)	137.43
FLAIR+RestoreFormer++ (Ours)	138.01

Table 6. Averaged runtime comparisons between FLAIR and other image DPM baselines for generating 10 frames. The experiments have been conducted on A100-80G for $4\times$ SR video JPEG restoration.

image DPM baseline DDNM for $4\times$ SR video JPEG restoration in Table 6. It is worth to note that, while we observe that FLAIR exhibits relatively slow processing speeds, one may easily combine FLAIR with existing sampling acceleration methods, such as staring from refined $\boldsymbol{x}_K$ [8], ODE based solvers [14, 15] and model distillation [18], etc.

3.2. Additional Visual Results

In Figs. 9 - 13, we present additional visual comparisons of several methods for video super-resolution on CelebV-Text and CelebV-HQ, where each row contains three frames. For each case, we also provide the zoomed-in region of the degraded inputs accordingly. In Figs. 14 - 16, we show more visual comparisons of several methods for video JPEG restoration with the zoomed-in regions. For video deblurring, we present the visual results through Fig.17 to 19. For real-world web video enhancement task, we assume the LQ inputs $\boldsymbol{y}$ corrupted by mixed degradation. Since our video DPM is trained for multi-variant degradation, we only need to fine-tune the data-consistency module. By fine-tuning the forward-model such that $\mathcal{A}\mathcal{A}^+\boldsymbol{y} \approx \boldsymbol{y}$, we observe that the degradation of $4\times$ SR with Gaussian kernel of width$= 1.6$, JPEG $Q = 90$ works the best. In Fig. 20, we present more visual results of our FLAIR compared with several baseline methods. It is worth to note that our method outperforms the baselines on real-world data even when the degradation is not known exactly. One can see from Fig. 20 that our designed two stage enhancement modules together can improve visual quality while preserving the data-consistency effectively.

4. Limitations and Societal Impacts

Although FLAIR achieves state-of-the-art performance in face video restoration, it still has some limitations. For example, the complexity of the reverse sampling step increases with respect to the conditional clip length and image spatial dimension. One possible solution is to develop a more efficient latent space diffusion for scale ability. As for societal impacts, similar to other restoration methods, FLAIR may bring privacy concerns after restoring low-quality face videos and lead to misjudgments if used for sensitive contents. One possible solution to mitigate this risk is to limit the usage of the model for sensitive or critical videos.

Figure 4. Visual comparisons of $4\times$ face video super-resolution with Gaussion blur kernel of width$=2$ and AWGN $\sigma = 0.05$ on CelebV-Text [25] (*top*) and CelebV-HQ [28] (*bottom*), respectively. Note the perceptual quality improvements of our FLAIR by applying different backbones for facial region enhancement. Best viewed by zooming in the display.

References

[1] G. Boracchi and A. Foi. Modeling the performance of image restoration from motion blur. *IEEE Trans. Image Process.*, 21(8):3502–3517, 2012. 1, 7

[2] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 3

[3] K. C. Chan, X. Wang, K. Yu, C. Dong, and C. C. Loy. Basicvsr: The search for essential components in video super-resolution and beyond. In *Proc. CVPR*, pages 4947–4956, 2021. 3

[4] K. C. Chan, S. Zhou, X. Xu, and C. C. Loy. Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. In *Proc. CVPR*, pages 5972–5981, 2022. 2, 3

Figure 5. Examples of synthetically generated random kernels, following [1, 26], used to generate the video deblurring dataset.

Figure 6. Visual comparisons of 16× face video super-resolution on CelebV-Text [25] (long clips). Note the perceptual quality and temporal coherent improvements of our FLAIR applying the combination of RFE and temporal attention module.

[5] C. Chen, X. Li, L. Yang, X. Lin, L. Zhang, and K. K. Wong. Progressive semantic-aware style transformation for blind face restoration. In *Proc. CVPR*, pages 11896–11905, 2021. 3

[6] Z. Chen, J. He, X. Lin, Y. Qiao, and C. Dong. Towards real-world video face restoration: A new benchmark. *arXiv preprint arXiv:2404.19500*, 2024. 4

[7] J. Choi, S. Kim, Y. Jeong, Y. Gwon, and S. Yoon. ILVR: Conditioning method for denoising diffusion probabilistic models. In *Proc. ICCV*, pages 14347–14356, 2021. 3, 5

[8] H. Chung, B. Sim, and J. C. Ye. Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. In *Proc. CVPR*, pages 12413–12422, 2022. 5

[9] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou. Retinaface: Single-stage dense face localisation in the wild. *arXiv:1905.00641*, 2019. 3

[10] Y. Gu, X. Wang, L. Xie, C. Dong, G. Li, Y. Shan, and M. Cheng. VQFR: Blind face restoration with vector-quantized dictionary and parallel decoder. In *Proc. ECCV*, pages 126–143. Springer, 2022. 3, 4, 5

[11] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *Proc. CVPR*, pages 16000–16009, 2022. 1

[12] J. Ho and T. Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 1

[13] Jingyun Liang, Jiezhang Cao, Yuchen Fan, Kai Zhang, Rakesh Ranjan, Yawei Li, Radu Timofte, and Luc Van Gool. VRT: A video restoration transformer. *arXiv:2201.12288*, 2022. 2, 3, 4

[14] L. Liu, Y. Ren, Z. Lin, and Z. Zhao. Pseudo numerical methods for diffusion models on manifolds. In *Proc. ICLR*, 2022. 5

Figure 7. Visual comparisons of *real-world* web video enhancement. Each row consists of three video frames, with an interval of around fifteen frames between each selected frame. Best viewed by zooming in the display.

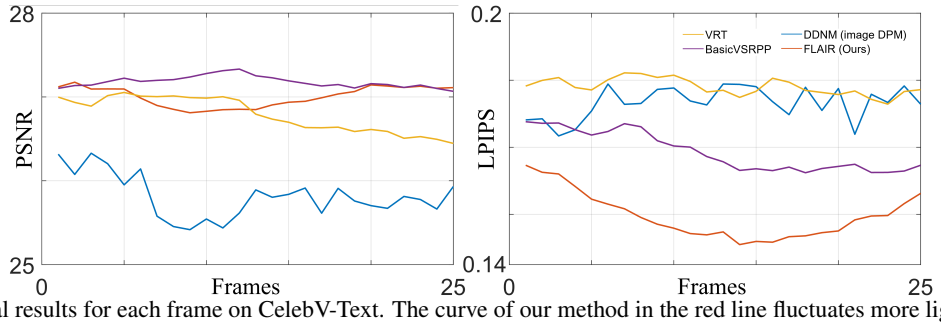

Figure 8. Numerical results for each frame on CelebV-Text. The curve of our method in the red line fluctuates more lightly than the image DPM in the blue line, matching general video SR methods (VRT/BasicVSRPP)

[15] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu. DPM-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Proc. NeurIPS*, 35:5775–5787, 2022. 5

[16] Alex Rogozhnikov. Einops: Clear and reliable tensor manipulations with einstein-like notation. In *International Conference on Learning Representations*, 2021. 1

[17] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton,

S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *Proc. NeurIPS*, 2022. 1

[18] T. Salimans and J. Ho. Progressive distillation for fast sampling of diffusion models. In *Proc. ICLR*, 2022. 5

[19] X. Wang, Y. Li, H. Zhang, and Y. Shan. Towards real-world blind face restoration with generative facial prior. In *Proc. CVPR*, pages 9168–9178, 2021. 3

[20] Y. Wang, J. Yu, and J. Zhang. Zero-shot image restoration using denoising diffusion null-space model. In *Proc. ICLR*, 2023. 3, 4, 5

[21] Z. Wang, J. Zhang, T. Chen, W. Wang, and P. Luo. Restoreformer++: Towards real-world blind face restoration from undegraded key-value pairs. *IEEE TPAMI*, 2023. 3, 4, 5

[22] Z. Wang, Z. Zhang, X. Zhang, H. Zheng, M. Zhou, Y. Zhang, and Y. Wang. DR2: Diffusion-based robust degradation remover for blind face restoration. In *Proc. CVPR*, pages 1704–1713, 2023. 3, 4, 5

[23] J. Whang, M. Delbracio, H. Talebi, C. Saharia, A. G. Dimakis, and P. Milanfar. Deblurring via stochastic refinement. In *Proc. CVPR*, pages 16293–16303, 2022. 3

[24] T. Yang, P. Ren, X. Xie, and L. Zhang. GAN prior embedded network for blind face restoration in the wild. In *Proc. CVPR*, pages 672–681, 2021. 3, 4

[25] J. Yu, H. Zhu, L. Jiang, C. C. Loy, W. Cai, and W. Wu. Celebv-text: A large-scale facial text-video dataset. In *Proc. CVPR*, pages 14805–14814, 2023. 3, 4, 5, 6, 7, 11, 12, 13, 15, 16, 17, 19, 20, 21

[26] K. Zhang, L. V. Gool, and R. Timofte. Deep unfolding network for image super-resolution. In *Proc. CVPR*, pages 3217–3226, Jun. 2020. 1, 7

[27] S. Zhou, K. Chan, C. Li, and C. C. Loy. Towards robust blind face restoration with codebook lookup transformer. *Proc. NeurIPS*, 35:30599–30611, 2022. 3, 4, 5

[28] H. Zhu, W. Wu, W. Zhu, L. Jiang, S. Tang, L. Zhang, Z. Liu, and C. C. Loy. CelebV-HQ: A large-scale video facial attributes dataset. In *Proc. ECCV*, pages 650–667. Springer, 2022. 3, 4, 5, 6, 14, 18

[29] Y. Zhu, K. Zhang, J. Liang, J. Cao, B. Wen, R. Timofte, and L. Van Gool. Denoising diffusion models for plug-and-play image restoration. In *Proc. CVPR Workshops*, pages 1219–1229, 2023. 3, 4

Hyperparameter	Bicubic 8 ×	Bicubic 16 ×	Gaussian Blur	Motion Blur	JPEG
Model Architecture					
Channels	64	64	128	128	128
# Resblocks	1	1	2	2	2
Attention Resolutions	(64, 32)	(64, 32)	(32, 16, 8)	(32, 16, 8)	(32, 16, 8)
RFE Resolutions	(512, 256)	(512, 256)	(512, 256)	(512, 256)	(512, 256)
Channel Multiplier	(1, 2, 4, 8, 16)	(1, 2, 4, 8, 16)	(0.5, 1, 1, 2, 2, 4, 4)	(0.5, 1, 1, 2, 2, 4, 4)	(0.5, 1, 1, 2, 2, 4, 4)
# Attention Heads	-	-	-	-	-
Head Channels	64	64	64	64	64
Temporal Attention Window Size	7	7	5	5	5
Diffusion Setup					
# Diffusion Steps	2000	2000	1000	1000	1000
Noise Schedule	Linear	Linear	Linear	Linear	Linear
β_1	1×10^{-6}	1×10^{-6}	1×10^{-4}	1×10^{-4}	1×10^{-4}
β_T	0.01	0.01	0.02	0.02	0.02
Image DPM Training					
Batch size	64	64	64	64	64
Learning Rate	1.5×10^{-4}	1.5×10^{-4}	1.5×10^{-4}	1.5×10^{-4}	1.5×10^{-4}
Weight Decay	0.05	0.05	0.05	0.05	0.05
# Samples	2M	2M	2M	2M	2M
EMA rate	0.9999	0.9999	0.9999	0.9999	0.9999
Video DPM Fine-tuning					
Batch size	4	4	4	4	4
Frame Length N	10	10	10	10	10
Learning Rate	1×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}
Weight Decay	0.05	0.05	0.05	0.05	0.05
# Samples	0.3M	0.3M	0.3M	0.3M	0.3M
EMA rate	-	-	-	-	-
Sampling					
$\|K\|$	25	100	100	65	40
ρ_t	0.85	0.85	0.25	0.35	0.5
w_τ	0.85	0.7	0.75	0.1	0.5
τ	5	5	5	5	5
ζ	-	-	1000	1000	1000

Table 7. Hyperparameters used in our FLAIR implementations.

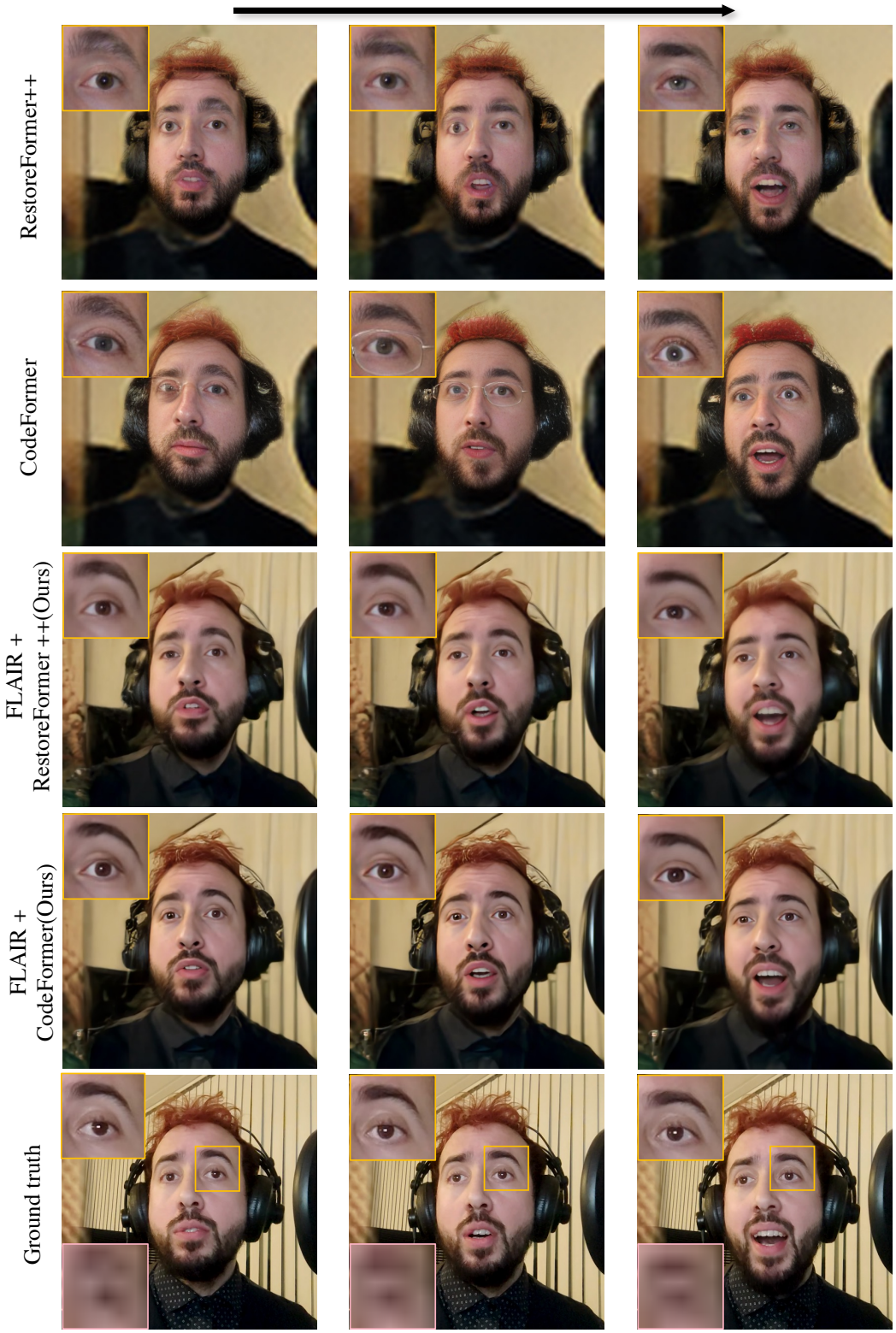

Figure 9. More visual results of 16× video super-resolution on CelebV-Text [25] dataset. Each row consists of three video frames, with an interval of five frames between each selected frame. The zoomed-in regions of each method are displayed in yellow boxes, along with their LQ counterparts in pink boxes. Best viewed by zooming in the display.

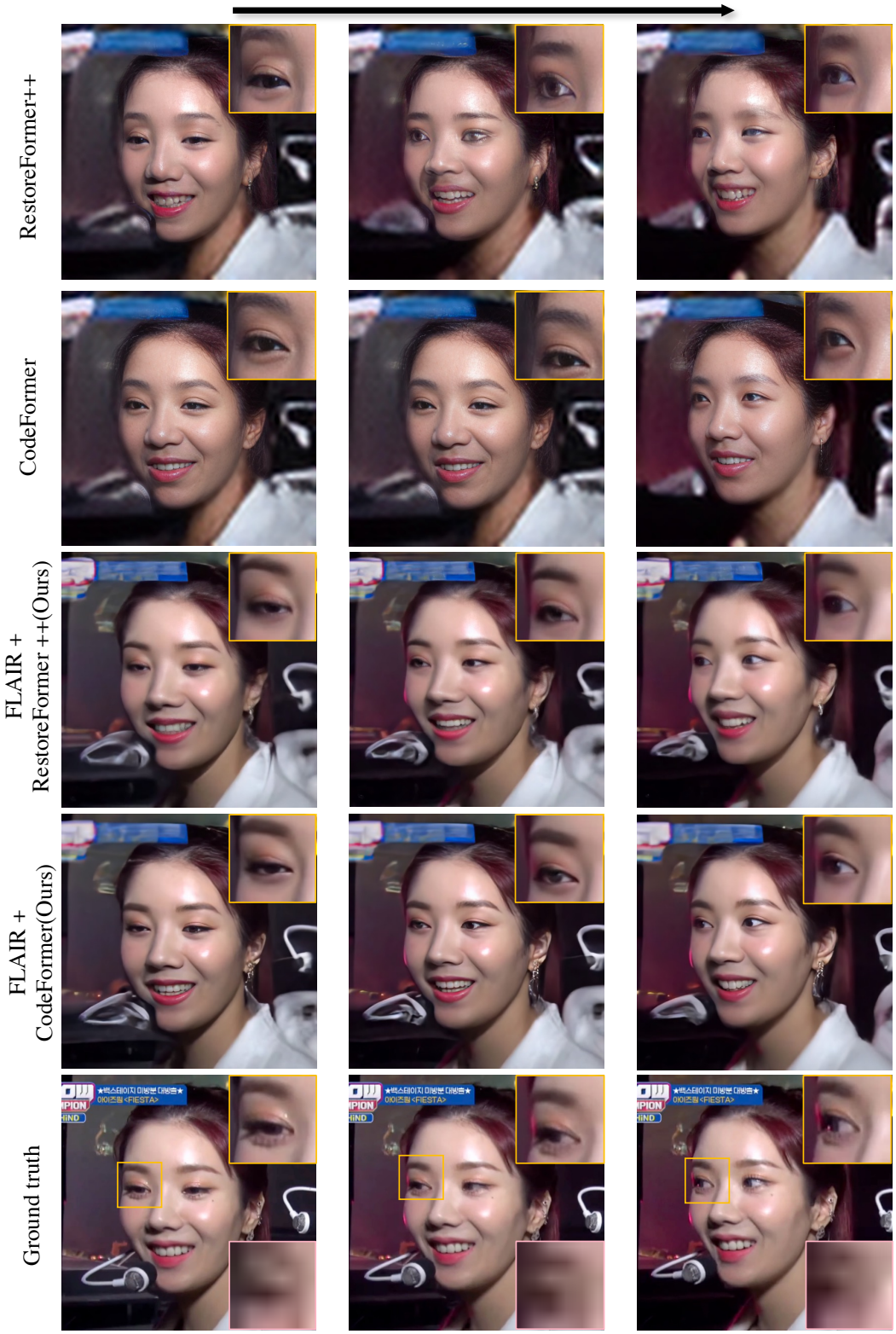

Figure 10. More visual results of 16× video super-resolution on CelebV-Text [25] dataset. Each row consists of three video frames, with an interval of five frames between each selected frame. The zoomed-in regions of each method are displayed in yellow boxes, along with their LQ counterparts in pink boxes. Best viewed by zooming in the display.

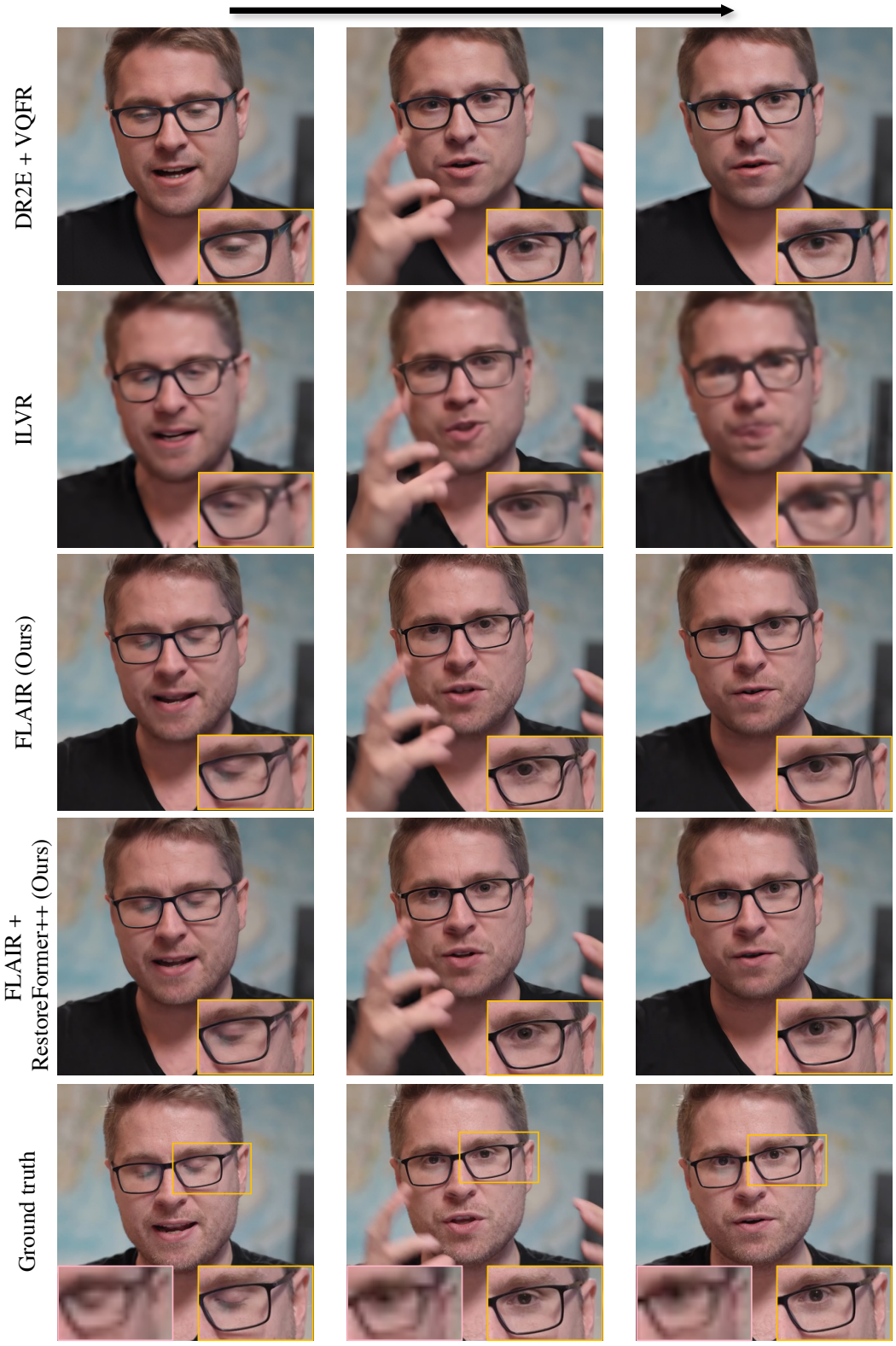

Figure 11. More visual comparisons of 8× face video super-resolution on CelebV-Text [25]. Each row consists of three video frames, with an interval of five frames between each selected frame. The zoomed-in regions of each method are displayed in yellow boxes, along with their LQ counterparts in pink boxes. Best viewed by zooming in the display.

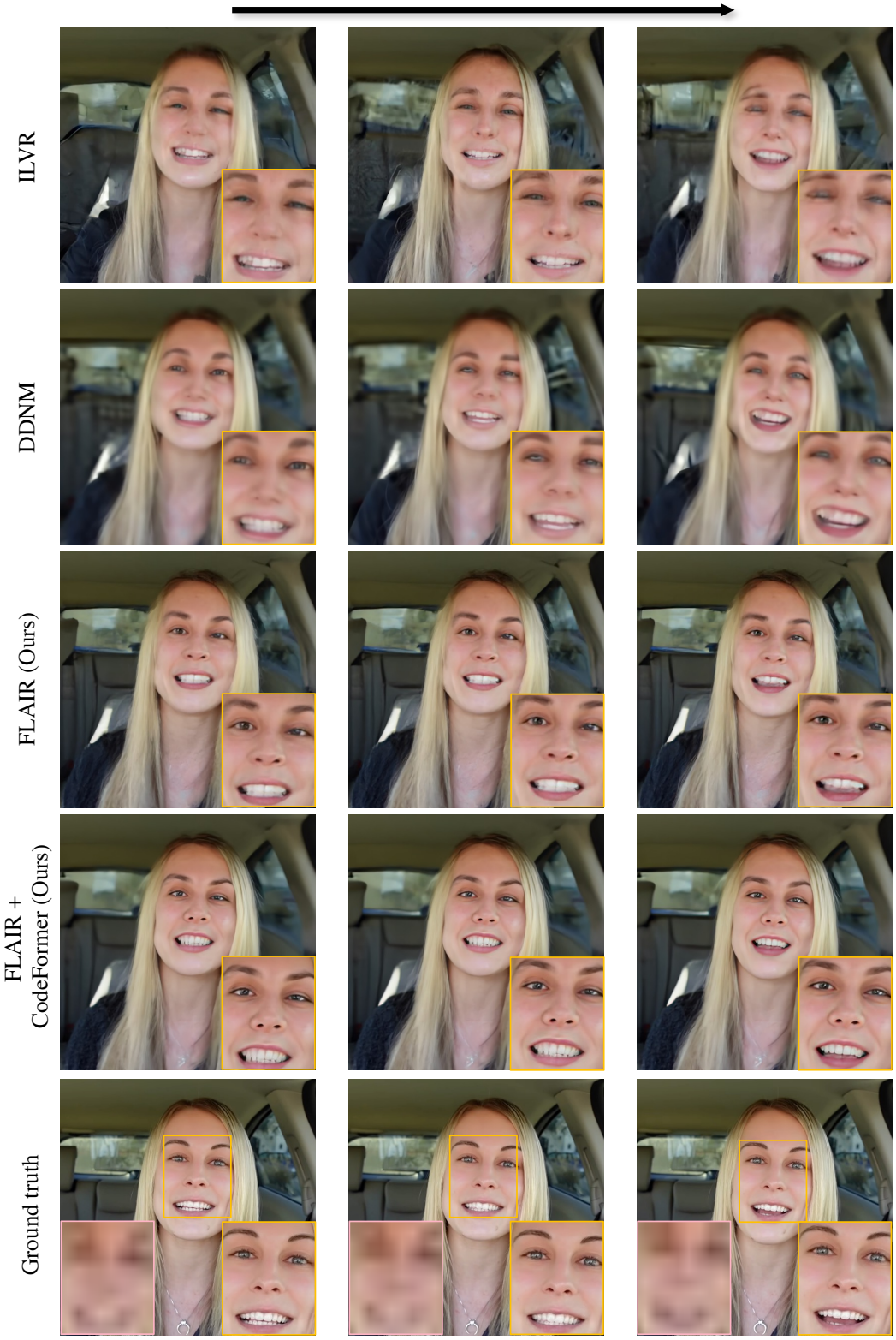

Figure 12. More visual comparisons of 16× face video super-resolution on CelebV-HQ [28]. Each row consists of three video frames, with an interval of five frames between each selected frame. The zoomed-in regions of each method are displayed in yellow boxes, along with their LQ counterparts in pink boxes. Best viewed by zooming in the display.

Figure 13. More visual comparisons of 16× face video super-resolution on CelebV-Text [25]. Each row consists of three video frames, with an interval of five frames between each selected frame. The zoomed-in regions of each method are displayed in yellow boxes, along with their LQ counterparts in pink boxes. Best viewed by zooming in the display.

Figure 14. More visual comparisons of $4\times$ face video JPEG restoration on CelebV-Text [25] dataset. Each row consists of three video frames, with an interval of five frames between each selected frame. The zoomed-in regions of each method are displayed in yellow and green boxes, along with their LQ counterparts in pink and blue boxes. Best viewed by zooming in the display.

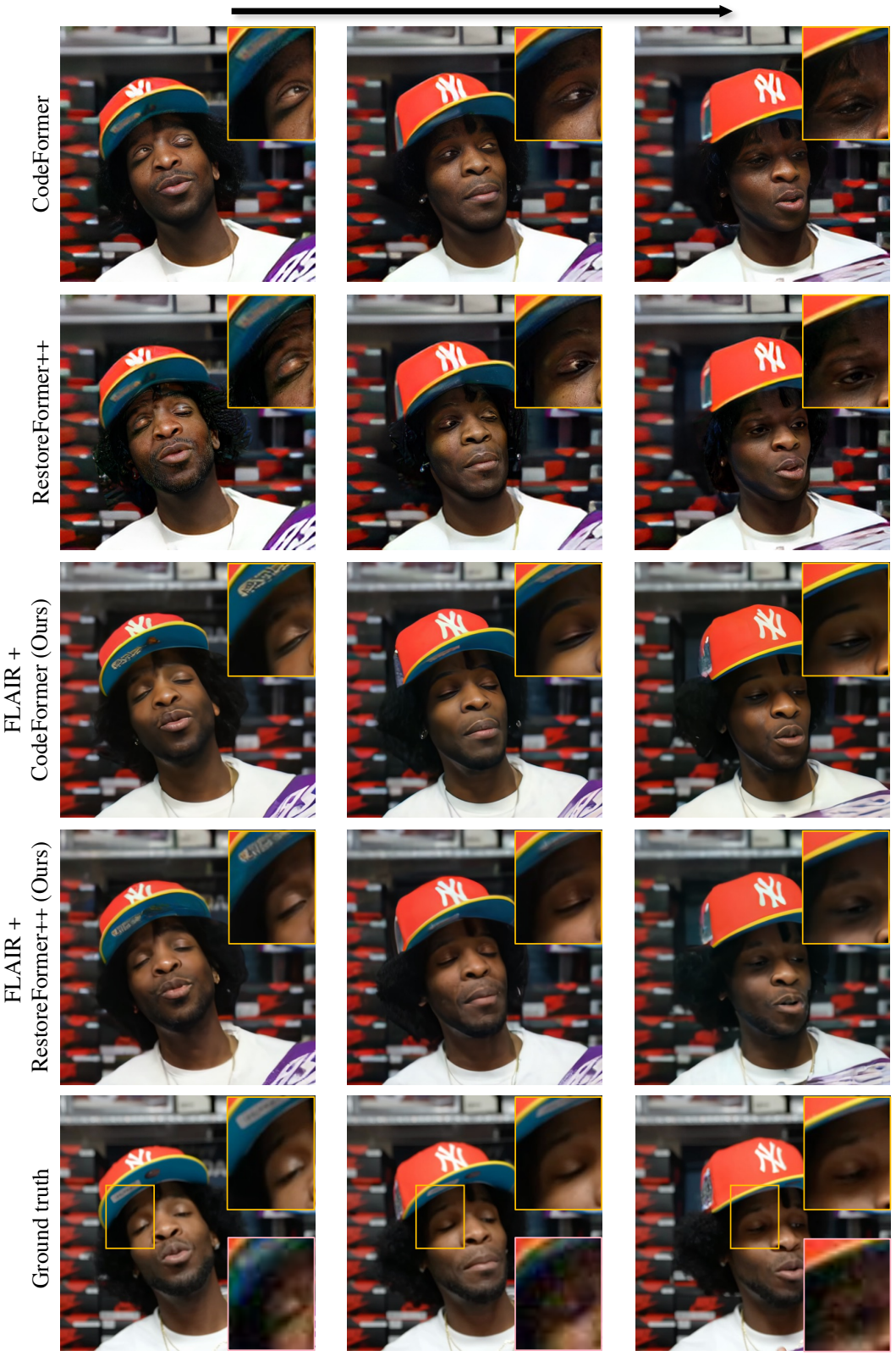

Figure 15. Visual comparisons of $4\times$ face video JPEG restoration on CelebV-Text [25]. Each row consists of three video frames, with an interval of five frames between each selected frame. The zoomed-in regions of each method are displayed in yellow boxes, along with their LQ counterparts in pink boxes. Best viewed by zooming in the display.

Figure 16. Visual comparisons of $4\times$ face video JPEG restoration on CelebV-HQ [28]. Each row consists of three video frames, with an interval of five frames between each selected frame. The zoomed-in regions of each method are displayed in yellow boxes. Best viewed by zooming in the display.

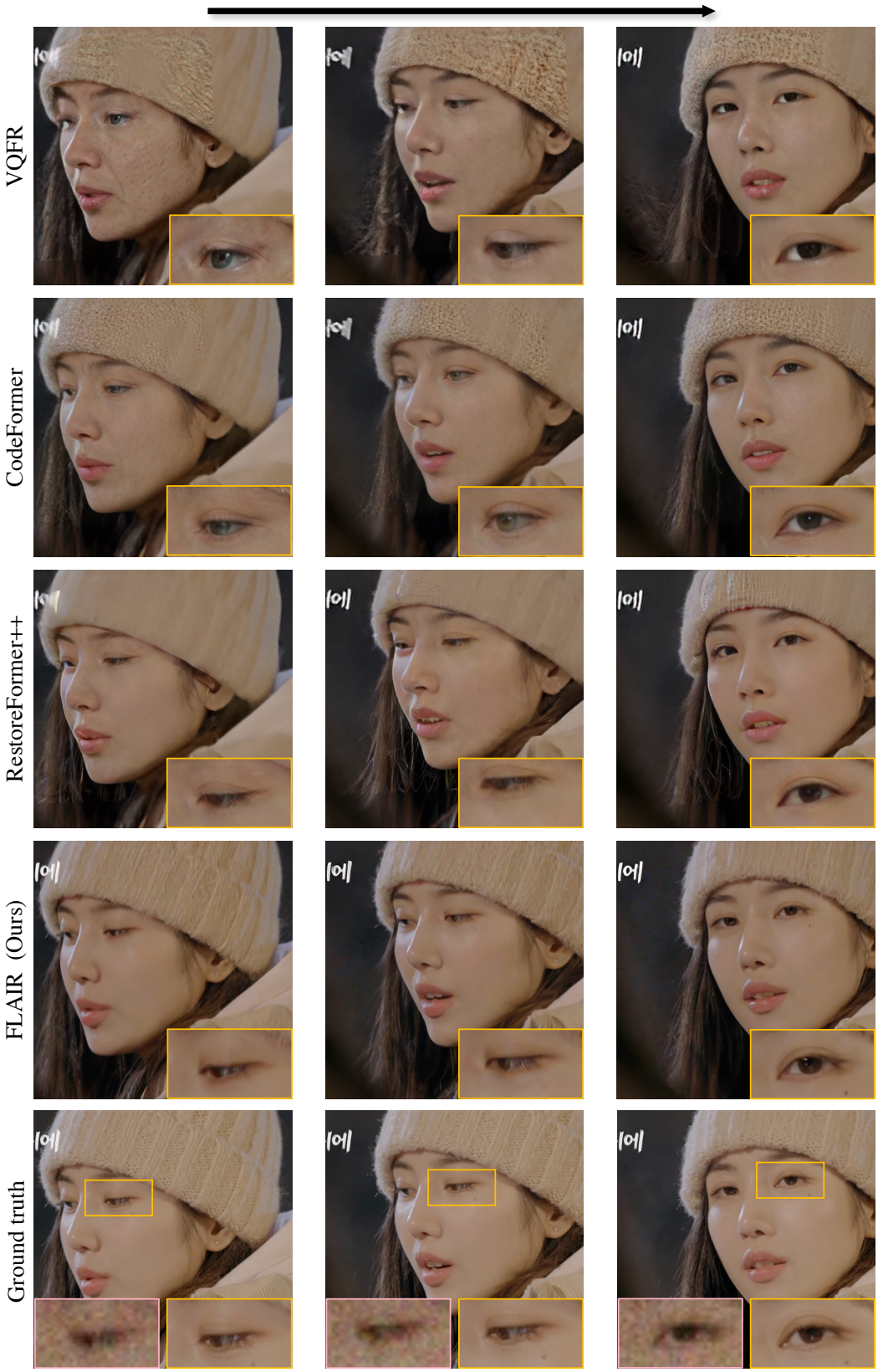

Figure 17. Visual comparisons of $4\times$ face video motion deblurring on CelebV-Text [25]. Each row consists of three video frames, with an interval of ten frames between each selected frame. The zoomed-in regions of each method are displayed in yellow boxes, along with their LQ counterparts in pink boxes. Best viewed by zooming in the display.

Figure 18. Visual comparisons of $4\times$ face video motion deblurring on CelebV-Text [25]. Each row consists of three video frames, with an interval of ten frames between each selected frame. The zoomed-in regions of each method are displayed in yellow boxes, along with their LQ counterparts in pink boxes. Best viewed by zooming in the display.

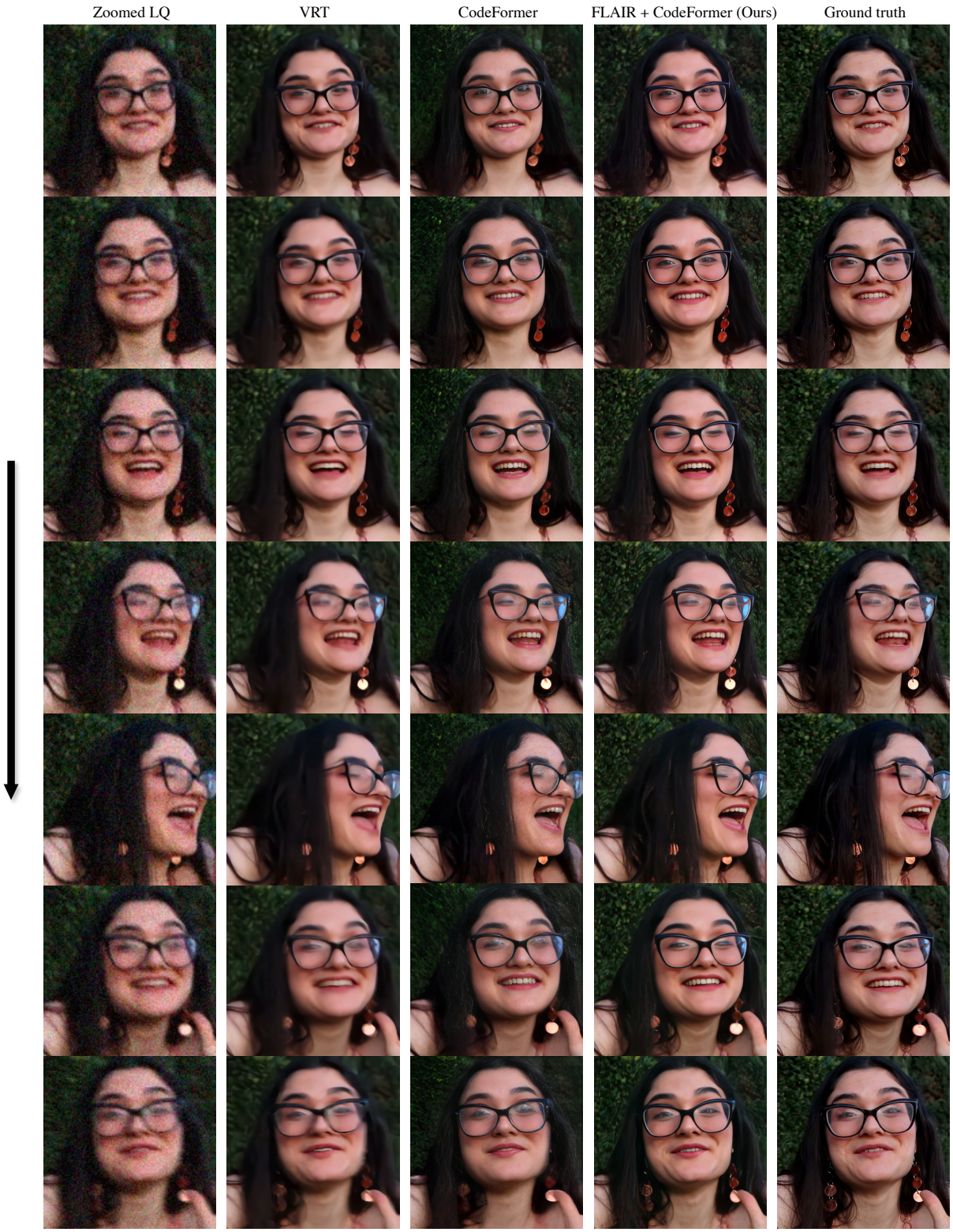

Figure 19. Visual comparisons of $4\times$ face video motion deblurring on CelebV-Text [25]. Each column consists of seven video frames, with an interval of ten frames between each selected frame. Best viewed by zooming in the display.

Figure 20. Visual comparisons of *real-world* web video enhancement. Each column consists of six video frames, with an interval of around fifteen frames between each selected frame. Best viewed by zooming in the display.