VISION-XL: High Definition Video Inverse Problem Solver using Latent Image Diffusion Models

Supplementary Material

6. Experimental details

6.1. Implementation of Comparative Methods

SVI [13]. For SVI, we use the official implementation[3]. Specifically, we utilize the same pre-trained image diffusion model, the unconditional ADM [6]. Following the protocol described in [13], we set the parameters as $l = 5$ and $\eta = 0.8$ with 100 NFE sampling. Since SVI officially supports a resolution of 256×256, we applied patch-based reconstruction to ensure fair comparisons at identical resolutions.

DiffIR2VR [33]. For DiffIR2VR, we use the official implementation[4]. Specifically, we employ the same pre-trained image diffusion model, Stable Diffusion 2.1 [19]. DiffIR2VR is designed to support only super-resolution (SR) within the scope of our inverse problem. Therefore, we conducted SR experiments exclusively. Following the protocol in [33], we set the upscale factor to 4 and the CFG scale factor to 4, with 50 NFE sampling. DiffIR2VR officially supports resolutions of 480×854. To ensure fair comparisons across resolutions, we applied patch-based reconstruction. For different aspect ratios, we set the resolution to 480×854 for landscape orientation, 854×480 for vertical orientation, and 512×512 for square.

ADMM-TV. Following the protocol in [13], we optimize the following objective:

$$\boldsymbol{X}^* = \underset{\boldsymbol{X}}{\operatorname{argmin}} \frac{1}{2}\|\mathcal{A}\boldsymbol{X} - \boldsymbol{Y}\|_2^2 + \lambda\|\boldsymbol{D}\boldsymbol{X}\|_1, \tag{10}$$

where $\boldsymbol{D} = [\boldsymbol{D}_t, \boldsymbol{D}_h, \boldsymbol{D}_w]$ corresponds to the classical Total Variation (TV) regularization. Here, t, h, and w represent temporal, height, and width directions, respectively. The outer iterations of ADMM were set to 30, and the inner iterations of conjugate gradient (CG) were set to 20, consistent with the settings in [13]. The parameters were set to $(\rho, \lambda) = (1, 0.001)$. The initial value of $\boldsymbol{X}$ was set to zero.

7. Extension to blind video inverse problems

Our method can be extended to address blind video inverse problems, such as blind video deblurring, demonstrated using the widely-used GoPro dataset [15]. Here, we provide an example application of our method to blind video deblurring, showing its potential as a general framework for solving blind video inverse problems.

Algorithm 2 Ours (blind) - Blind video deconvolution

```
Require: $\mathcal{E}_\theta^{(t)}, \boldsymbol{E}_\theta, \boldsymbol{D}_\theta, \boldsymbol{Y}, \tau, l, \sigma_t, \{\bar{\alpha}_t\}_{t=1}^T, f_\phi$
$\boldsymbol{X}_{\text{pre}} \leftarrow f_\phi(\boldsymbol{Y})$    ▷ Round 1 with estimated PSF
$\boldsymbol{h}_\sigma \leftarrow \arg\min_{\boldsymbol{h}_\sigma} \|\boldsymbol{Y} - \boldsymbol{X}_{\text{pre}} * \boldsymbol{h}_\sigma\|^2$
$\boldsymbol{z}_0 \leftarrow \boldsymbol{E}_\theta(\boldsymbol{Y})$
$\boldsymbol{z}_\tau \leftarrow \text{DDIM}^{-1}(\boldsymbol{z}_0)$
for $t = \tau : 2$ do
    $\hat{\boldsymbol{z}}_t \leftarrow \left(\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\mathcal{E}_\theta^{(t)}(\boldsymbol{z}_t)\right)/\sqrt{\bar{\alpha}_t}$
    $\hat{\boldsymbol{X}}_t \leftarrow \boldsymbol{D}_\theta(\hat{\boldsymbol{z}}_t)$
    $\bar{\boldsymbol{X}}_t := \arg\min_{\boldsymbol{X} \in \hat{\boldsymbol{X}}_t + \mathcal{K}_l} \|\boldsymbol{Y} - \boldsymbol{X} * \boldsymbol{h}_\sigma\|^2$
    $\bar{\boldsymbol{X}}_t \leftarrow \bar{\boldsymbol{X}}_t * \boldsymbol{h}_{\sigma_t}$
    $\bar{\boldsymbol{z}}_t = \boldsymbol{E}_\theta(\bar{\boldsymbol{X}}_t)$
    $\boldsymbol{z}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\bar{\boldsymbol{z}}_t + \sqrt{1-\bar{\alpha}_{t-1}}\mathcal{E}_t$
end for
$\boldsymbol{z}_0 \leftarrow \left(\boldsymbol{z}_1 - \sqrt{1-\bar{\alpha}_1}\mathcal{E}_\theta^{(1)}(\boldsymbol{z}_1)\right)/\sqrt{\bar{\alpha}_1}$
$\boldsymbol{h}_\sigma \leftarrow \arg\min_{\boldsymbol{h}_\sigma} \|\boldsymbol{Y} - \boldsymbol{D}_\theta(\boldsymbol{z}_0) * \boldsymbol{h}_\sigma\|^2$    ▷ Round 2 with refined PSF
for $t = \tau : 2$ do
    $\hat{\boldsymbol{z}}_t \leftarrow \left(\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\mathcal{E}_\theta^{(t)}(\boldsymbol{z}_t)\right)/\sqrt{\bar{\alpha}_t}$
    $\hat{\boldsymbol{X}}_t \leftarrow \boldsymbol{D}_\theta(\hat{\boldsymbol{z}}_t)$
    $\bar{\boldsymbol{X}}_t := \arg\min_{\boldsymbol{X} \in \hat{\boldsymbol{X}}_t + \mathcal{K}_l} \|\boldsymbol{Y} - \boldsymbol{X} * \boldsymbol{h}_\sigma\|^2$
    $\bar{\boldsymbol{X}}_t \leftarrow \bar{\boldsymbol{X}}_t * \boldsymbol{h}_{\sigma_t}$
    $\bar{\boldsymbol{z}}_t = \boldsymbol{E}_\theta(\bar{\boldsymbol{X}}_t)$
    $\boldsymbol{z}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\bar{\boldsymbol{z}}_t + \sqrt{1-\bar{\alpha}_{t-1}}\mathcal{E}_t$
end for
$\boldsymbol{z}_0 \leftarrow \left(\boldsymbol{z}_1 - \sqrt{1-\bar{\alpha}_1}\mathcal{E}_\theta^{(1)}(\boldsymbol{z}_1)\right)/\sqrt{\bar{\alpha}_1}$
return $\boldsymbol{z}_0$
```

In the context of blind deconvolution, an intuitive strategy is to alternate between point spread function (PSF) estimation and deconvolution. Since accurately estimating the initial PSF is challenging, we first employ a lightweight video deblurring module, DeepDeblur [15], for preliminary restoration. The initial PSF is then estimated based on this pre-restored video. Using the estimated PSF, we perform a Round 1 reconstruction with our proposed method. Subsequently, the PSF is refined based on the output of this reconstruction. The refined PSF is then utilized for the final (Round 2) reconstruction, yielding an improved result.

In summary, our method incorporates a lightweight pre-restoration step to estimate the initial PSF and employs a two-round reconstruction pipeline to achieve high-quality restoration through PSF refinement. The detailed steps of the algorithm are outlined in Algorithm 2.

[3] https://github.com/solving-video-inverse/codes

[4] https://github.com/jimmycv07/DiffIR2VR-Zero

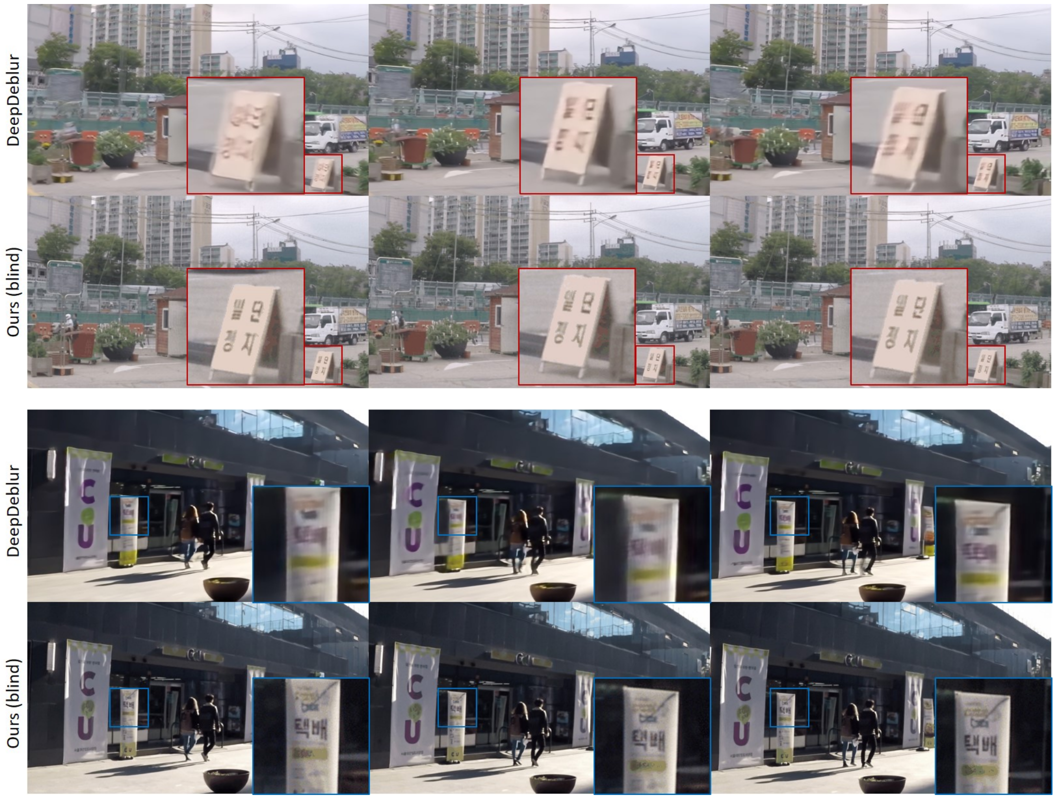

Figure 8. Qualitative comparison of video deblurring results on the GoPro test dataset [15] compared with DeepDeblur [15].

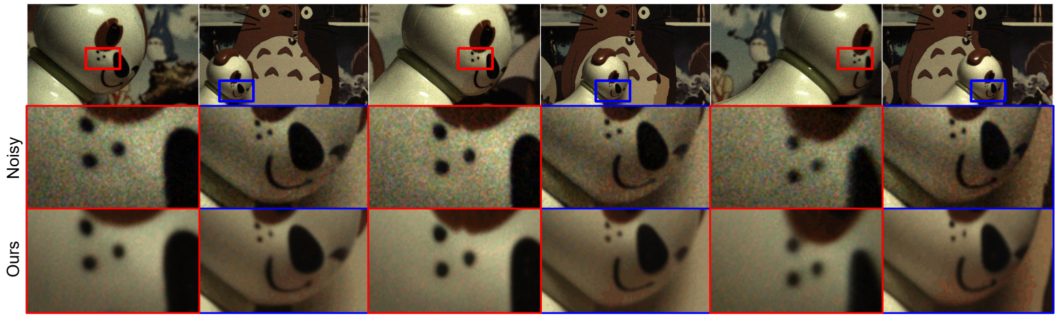

Figure 9. Qualitative evaluation of real-world video denoising results on the CRVD dataset [34]. Our method robustly removes real noise across different ISO settings.

The GoPro dataset consists of 240 fps videos captured with a GoPro camera, where motion blur is synthetically generated by averaging 7 to 13 consecutive frames [15]. For our experiments, we used the GoPro test dataset and performed blind video reconstruction using Algorithm 2, generating blurred inputs by randomly averaging 7 to 13 frames. To evaluate the effectiveness of our approach, we compared our reconstruction results with those from the pre-restoration module. Our method significantly improves reconstruction quality, yielding highly detailed results. As shown in Fig. 8, zoomed-in views of signboards and billboards reveal that our method recovers fine details, such as text, with greater precision. This improvement demonstrates how incorporating a diffusion prior enables more accurate PSF estimation. Additionally, it highlights the potential of our method to extend to various blind inverse problems.

Furthermore, we validated our method on the CRVD dataset [34], which contains real-world noisy videos captured at various ISO levels. By employing the extended CG step for noisy restorations [4], our method readily adapts to real-world video denoising. As shown in Fig. 9, it robustly removes noise across different ISO settings.

8. Comprehensive visualizations

For an in-depth understanding of the experimental results, we provide video visualizations on our anonymous project page[5]. The page features 36 paired visualizations of measurements and reconstructions across various aspect ratios and degradation types. As shown on the project page, our method delivers highly satisfactory reconstruction results for various spatio-temporal inverse problems.

Additional comparisons with baselines are available on our supplementary anonymous project page[6]. In baseline comparisons, ADMM-TV struggles to reconstruct temporal degradations, and SVI [13] exhibits poor temporal consistency. DiffIR2VR [33] frequently fails to reconstruct and produces undesired artifacts, likely due to errors in the optical flow estimation module. In contrast, our approach achieves superior performance across various spatio-temporal inverse problems.

We also provide visualizations of ablation studies. Regarding initialization effects, our pseudo-batch inversion significantly improves temporal consistency compared to random noise initialization or batch-consistent noise initialization [13]. Regarding the low-pass filter effect, we observe that applying a well-scheduled low-pass filter produces cleaner results with fewer artifacts. Without the low-pass filter, artifacts such as the grid pattern under the red bridge or the lattice-like texture on the body of the sea snake are noticeable.

We strongly encourage you to visit these project pages to explore the superior reconstruction performance of our method.

[5] https://vision-xl.github.io/

[6] https://vision-xl.github.io/supple/