# SUPPLEMENTARY: PRACTICAL REAL VIDEO DENOISING WITH REALISTIC DEGRADATION MODEL

**Anonymous authors**
Paper under double-blind review

**Organization.** We organize our supplementary materials as follows. For the theory part, we provide detailed proofs of the theorems in Section A. In Sections B, we provide more detailed formulations of spatial and temporal denoising. In Section C, we provide detailed settings of noise degradation in the experiments. In Section D, we provide more settings, details and results of the experiments. In Section E, we give the limitations and societal impacts of our proposed method.

## A THEORETICAL ANALYSIS

We build a relationship between the denoising problem and general optimization problem training with noise. Given a training data $(\boldsymbol{x}, \boldsymbol{y})$, the general optimization problem training with some kind of noise $\boldsymbol{z}_\sigma$ can be written as:

$$\min_f \mathbb{E}_\sigma \left[ \mathbb{E}_{(\boldsymbol{x}_\sigma, \boldsymbol{y})} \left[ \|f(\boldsymbol{x}_\sigma) - \boldsymbol{y}\|^2 \right] \right]. \tag{1}$$

Based on the analysis of (Bishop, 1995), we first provide the following theorem.

**Theorem 1 (Effect of noise degradations)** *Let $\boldsymbol{z}_\sigma = g(\boldsymbol{x}) - \boldsymbol{x}$, and assume that the mean and variance of the noise distribution are $0$ and $\eta^2(\boldsymbol{z}_\sigma)$, then the loss (1) , i.e.,*

$$\begin{aligned}\mathbb{E}_{\boldsymbol{z}_\sigma} \left[ \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})} \left[ \|f(\boldsymbol{x}+\boldsymbol{z}_\sigma) - \boldsymbol{y}\|^2 \right] \right] &= \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})} \left[ \|f(\boldsymbol{x}) - \boldsymbol{y}\|^2 \right] \\ &+ \eta^2(\boldsymbol{z}_\sigma) \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})} \left[ \left\| \frac{\partial f}{\partial \boldsymbol{x}} \right\|^2 + \frac{1}{2} \left( f(\boldsymbol{x}) - \boldsymbol{y} \right)^\top \frac{\partial^2 f}{\partial \boldsymbol{x}^2} \mathbf{1} \right].\end{aligned} \tag{2}$$

**Proof** Based on the expectation *w.r.t.* $\boldsymbol{x}, \boldsymbol{y}$ and $\boldsymbol{z}_\sigma$, we have

$$\mathbb{E}_{\boldsymbol{z}_\sigma} \left[ \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})} \left[ \|f(\boldsymbol{x}+\boldsymbol{z}_\sigma) - \boldsymbol{y}\|^2 \right] \right] \tag{3}$$

$$= \int\int\int \|f(\boldsymbol{x}+\boldsymbol{z}_\sigma) - \boldsymbol{y}\|^2 p(\boldsymbol{x}) p(\boldsymbol{y}|\boldsymbol{x}) p(\boldsymbol{z}_\sigma) d\boldsymbol{x} d\boldsymbol{y} d\boldsymbol{z}_\sigma \tag{4}$$

$$= \int\int\int \sum_k (f_k(\boldsymbol{x}+\boldsymbol{z}_\sigma) - y_k)^2 p(\boldsymbol{x}) p(\boldsymbol{y}|\boldsymbol{x}) p(\boldsymbol{z}_\sigma) d\boldsymbol{x} d\boldsymbol{y} d\boldsymbol{z}_\sigma \tag{5}$$

$$= \int\int \sum_k (f_k(\boldsymbol{x}) - y_k)^2 p(\boldsymbol{x}) p(\boldsymbol{y}|\boldsymbol{x}) d\boldsymbol{x} d\boldsymbol{y} \tag{6}$$

$$+ \int\int \sum_{i,k} \left[ \left( \frac{\partial f_k}{\partial x_i} \right)^2 + \frac{1}{2} (f_k(\boldsymbol{x}) - y_k) \frac{\partial^2 f_k}{\partial x_i^2} \right] p(\boldsymbol{x}) p(\boldsymbol{y}|\boldsymbol{x}) d\boldsymbol{x} d\boldsymbol{y} \tag{7}$$

$$= \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})} \left[ \|f(\boldsymbol{x}) - \boldsymbol{y}\|^2 \right] + \eta^2(\boldsymbol{z}_\sigma) \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})} \left[ \left\| \frac{\partial f}{\partial \boldsymbol{x}} \right\|^2 + \frac{1}{2} \left( f(\boldsymbol{x}) - \boldsymbol{y} \right)^\top \frac{\partial^2 f}{\partial \boldsymbol{x}^2} \mathbf{1} \right], \tag{8}$$

where the Equations (6-7) hold the assumption of the noise $\boldsymbol{z}_\sigma$, *i.e.*,

$$\int z_i p(\boldsymbol{z}_\sigma) d\boldsymbol{z}_\sigma = 0, \quad \int z_i z_j p(\boldsymbol{z}_\sigma) d\boldsymbol{z}_\sigma = \eta^2(\boldsymbol{z}_\sigma) \delta_{ij} \tag{9}$$

and use the Taylor series of the noise $\boldsymbol{z}_\sigma$, *i.e.*,

$$f_k(\boldsymbol{x}+\boldsymbol{z}_\sigma) = f_k(\boldsymbol{x}) + \sum_i z_i \left. \frac{\partial f_k}{\partial x_i} \right|_{\boldsymbol{z}_\sigma = \mathbf{0}} + \frac{1}{2} \sum_i \sum_j z_i z_j \left. \frac{\partial^2 f_k}{\partial x_i \partial x_j} \right|_{\boldsymbol{z}_\sigma = \mathbf{0}} + \mathcal{O}(\boldsymbol{z}_\sigma^3). \tag{10}$$

□

**Note that when $x$=$y$, the general learning problem turns to a problem of learning AutoEncoder.** Based on Theorem 1, we have rewrite the following theorem when $\boldsymbol{x}$=$\boldsymbol{y}$.

**Theorem 1** ***(Effect of noise degradations)*** *Let $\boldsymbol{z}_\sigma = g(\boldsymbol{x}) - \boldsymbol{x}$, and assume that the mean and variance of the noise distribution are $0$ and $\eta^2(\boldsymbol{z}_\sigma)$, then the loss (1) , i.e.,*

$$\mathbb{E}_\sigma\left[\mathbb{E}_{(\boldsymbol{x})}\left[\|f(\boldsymbol{x}_\sigma)-\boldsymbol{x}\|^2\right]\right] = \mathbb{E}_{\boldsymbol{x}}\left[\|f(\boldsymbol{x})-\boldsymbol{x}\|^2\right] + \eta^2(\boldsymbol{z}_\sigma)\mathbb{E}_{\boldsymbol{x}}\left[\left\|\frac{\partial f}{\partial \boldsymbol{x}}\right\|^2 + \frac{1}{2}\left(f(\boldsymbol{x})-\boldsymbol{x}\right)^\top \frac{\partial^2 f}{\partial \boldsymbol{x}^2}\mathbf{1}\right]. \tag{11}$$

**Proof** Let $\boldsymbol{x} = \boldsymbol{y}$ in Theorem 1, we complete the proof. □

From this theorem, the loss (1) trained with our noise degradations is equivalent to a Autoencoder loss with a regularization term. The parameter $\eta^2(\boldsymbol{z}_\sigma)$ is related to the amplitude or variance of the noise $\boldsymbol{z}_\sigma$ and controls how the regularization term influences the loss.

## B More Details of Spatial and Temporal Denoising

**Spatial denoising.** Given a feature, we use multi-layered residual blocks (He et al., 2016) to implement the spatial encoder $E_{\text{spatial}}$ to extract deep features and reduce the spatial noise at each scale, *i.e.*,

$$E_{\text{spatial}}\left(\boldsymbol{g}_i^{s-1}\right) = R_N \circ \cdots \circ R_1\left(\boldsymbol{g}_i^{s-1}\right), \tag{12}$$

where $\circ$ is a function composition, and each $R_i$ is a residual block. In the experiment, we set $N = 5$ and the number of features channels is 64. Given a feature $\boldsymbol{g}$, the residual block is formulated as

$$R_i(\boldsymbol{g}) = \boldsymbol{g} + \mathrm{Conv}_2(\mathrm{ReLU}(\mathrm{Conv}_1(\boldsymbol{g}))), \tag{13}$$

where $\mathrm{Conv}_1$ and $\mathrm{Conv}_2$ are convolutional layers, and $\mathrm{ReLU}$ is an activation.

**Temporal denoising.** We implement $E_{\text{temporal}}$ by using the architecture of the flow-guided deformable alignment of (Chan et al., 2022a) to predict offset and mask in DCN (Zhu et al., 2019). Given denoised spatial features $\boldsymbol{g}_i^s$, we use the optical-flow-guided deformable alignment as our temporal encoder $E_{\text{temporal}}$ to compute the features at the $j$-th branch, *i.e.*,

$$\widehat{\boldsymbol{f}}_i = E_{\text{temporal}}\left(\boldsymbol{g}_i^s, \boldsymbol{f}_{i-1,j}^s, \boldsymbol{f}_{i-2,j}^s, \boldsymbol{o}_{i\to i-1}^s, \boldsymbol{o}_{i\to i-2}^s\right) \tag{14}$$

$$= \mathrm{DCN}([\boldsymbol{f}_{i-1}; \boldsymbol{f}_{i-2}], [\widetilde{\boldsymbol{o}}_{i\to i-1}; \widetilde{\boldsymbol{o}}_{i\to i-2}], [\boldsymbol{m}_{i\to i-1}; \boldsymbol{m}_{i\to i-2}]), \tag{15}$$

where the offsets and masks are formulated as

$$\widetilde{\boldsymbol{o}}_{i\to i-p} = \boldsymbol{o}_{i\to i-p} + \mathrm{Conv}([\boldsymbol{g}_i; \bar{\boldsymbol{f}}_{i-1}; \bar{\boldsymbol{f}}_{i-2}]), \tag{16}$$

$$\boldsymbol{m}_{i\to i-p} = \mathrm{Sigmoid}(\mathrm{Conv}([\boldsymbol{g}_i; \bar{\boldsymbol{f}}_{i-1}; \bar{\boldsymbol{f}}_{i-2}])), \tag{17}$$

where $p = 1, 2$ and $\boldsymbol{f}_{i-1}$ is a warped feature using the optical flow $\boldsymbol{o}_{i\to i-1}$, *i.e.*,

$$\bar{\boldsymbol{f}}_{i-1} = \mathrm{warp}(\boldsymbol{f}_{i-1}, \boldsymbol{o}_{i\to i-1}) \quad \text{and} \quad \bar{\boldsymbol{f}}_{i-2} = \mathrm{warp}(\boldsymbol{f}_{i-2}, \boldsymbol{o}_{i\to i-2}), \tag{18}$$

where $\mathrm{warp}(\cdot)$ is a warp function according to the optical flow. After reducing the temporal noise, we use another spatial encoder $E'_{\text{spatial}}$ with 7 residual blocks.

## C Experiment Details of Noise Degradation

**Noise.** In the experiment, we consider 6 kinds of noises in the degradations, including Gaussian noise, Poisson noise, Speckle noise, Processed camera sensor noise, JPEG compression noise and video compression noise. To explore the properties of video denoising, we use the default order of the following noise in Figure 1 (a) and 10 (Top).

- *Gaussian noise.* We uniformly sample noise levels $\sigma$ from $[2, 50]$. We randomly choose AWGN and grayscale AWGN with the probabilities of 0.6 and 0.4, respectively.
- *Poisson noise.* We add Poisson noise in color and grayscale images by sampling different noise levels. We first multiply the clean video by $10^\alpha$ in the function of Poisson distribution, where $\alpha$ is unformly chosen from $[2, 4]$ and divide by $10^\alpha$.
- *Speckle noise.* We sample the level of this noise from $[0, 50]$.

- *Processed camera sensor noise.* Inspired by (Zhang et al., 2022), the reverse ISP pipeline first get the raw image from an RGB image, then the forward pipeline constructs noisy raw image by adding noise to the raw image.
- *JPEG compression noise.* The JPEG quality factor is uniformly chosen from $[30, 95]$. JPEG compression noise will introduce $8 \times 8$ blocking artifacts.
- *Video compression noise.* We use the Pythonic operator av in FFmpeg to produce compression noise. We randomly selected codecs from ['libx264', 'h264', 'mpeg4'] and bitrate from [1e4, 1e5] during training.

**Blur.** In addition to noise, most real-world videos inherently suffer from blur structure in a digital camera. Thus, we consider two blur degradations, including Gaussian blur and resizing blur.

- *Gaussian blur.* We synthesize Gaussian blur with different kernels, including ['iso', 'aniso', 'generalized_iso', 'generalized_aniso', 'plateau_iso', 'plateau_aniso', 'sinc']. We randomly choose these kernels with the probabilities of [0.405, 0.225, 0.108, 0.027, 0.108, 0.027, 0.1]. The settings of these blur are the same as (Chan et al., 2022b).
- *Resizing blur.* We randomly draw the resize scales from [0.5, 2], and choose the interpolation mode from ['bilinear', 'area', 'bicubic'] with the same probability of $1/3$.

# D More Experiments

## D.1 More Details of Experiment Setting

We adopt Adam optimizer (Kingma & Ba, 2015) and Cosine Annealing scheme (Loshchilov & Hutter, 2016) to decay the learning rate from $1\times10^{-4}$ to $10^{-7}$. The patch size is $256\times256$, and batch size is 8. The number of input frames is 15. All experiments are implemented by PyTorch 1.9.1. We train a denoising model on 8 A100 GPUs. We use the pre-trained SPyNet (Ranjan & Black, 2017) to estimate the flow. Note that we fix the parameters of SPyNet during the training. We train our video denoiser with 150k iterations. For the synthetic Gaussian denoising, the learning rate of the generator is $1\times10^{-4}$. For real-world video denoising, the learning rates of the generator and discriminator are set to $5\times10^{-5}$ and $1\times10^{-4}$. The architecture of the generator is introduced in Section B. The architecture of the discriminator is the same as Real-ESRGAN (Wang et al., 2021). When training classic video denoising, we use Charbonnier loss (Charbonnier et al., 1994) due to its stability and good performance. For real video denoising, we first use Charbonnier loss to train a model, then we finetune the network by using the perceptual loss $\mathcal{L}_{pix}$ (Johnson et al., 2016) and adversarial loss $\mathcal{L}_{adv}$ (Goodfellow et al., 2014), *i.e.*, $\mathcal{L} = \mathcal{L}_{pix} + \lambda_1\mathcal{L}_{per} + \lambda_2\mathcal{L}_{adv}$, where $\lambda_1 = 1$ and $\lambda_2 = 5 \times 10^{-1}$. Code will be made publicly available.

## D.2 Training Loss and PSNR

To demonstrate the efficiency of our model, we show the training loss and PSNR, as shown in Figure 1. At every 10K iterations, the PSNR value is calculated on Set8 with the noise level of 10. The total training iterations is 150k and takes 3 days. The training loss decreases rapidly at early iterations and stay steady in the later iterations. The PSNR values on Set8 increase during the training. These results demonstrate that our model is easy to train to have good performance.

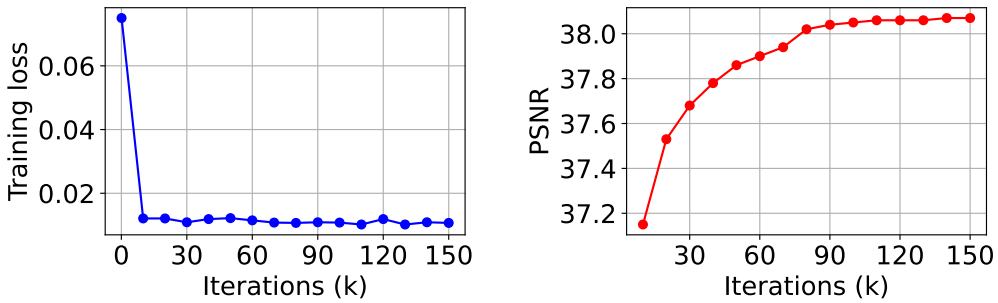


Figure 1: An illustration of training loss and PSNR.

### D.3 Differences of Image and Video Degradations

Our video degradation significantly differs from the existing single image degradation. First, we consider the blur degradation (Gaussian blur and resizing blur) which would change the statistics of other noises and make the noise more complex. Second, we consider different video compression noises which usually require temporal information for better noise removal. From Table 1, training without blur degradation and video compression noise lead to inferior performance, which demonstrates the dominant role.

Table 1: Ablation study on noise types on VideoLQ.

| Types | NIQE | BRISQUE | PIQE |
|---|---|---|---|
| w/o Blur degradation | 4.1643 | 34.8137 | 50.2962 |
| w/o video compression noise | 4.0537 | 31.8712 | 50.7835 |
| **Ours** | **4.0205** | **29.0212** | **45.0768** |

### D.4 Parameters for Noise Types.

We determine the parameters for each noise type according to the common well-studied settings or experimental analysis. For example, some parameter settings in image-based denoising methods (Zhang et al., 2022; 2021) have been well-studied. We further conduct an analysis for different parameters of noise degradation. Here, we analyze the bitrate range of video compression noise due to its importance in our noise degradation in Table 2. The model achieves the best performance with a bitrate range of $[1e4, 1e5]$, which accords with the setting in (Chan et al., 2022b).

Table 2: Performance of different bitrate ranges on VideoLQ.

| Types | NIQE | BRISQUE | PIQE |
|---|---|---|---|
| $[1e3, 1e4]$ | 4.2317 | 30.1674 | 46.7984 |
| $[1e4, 1e5]$ | **4.0205** | **29.0212** | **45.0768** |
| $[1e5, 1e6]$ | 4.1276 | 31.3297 | 49.4122 |

Actually, the included speckle noise is already a type of spatially correlated noise. For certain unseen applications, there may exist other types of noises. Without prior knowledge, it is difficult to cover all types of unseen noises. Thus, this paper considers the most common and general noises in our degradation model. Certainly, if a noise type is dominant for a certain application, one can augment it into our degradation model to match the noise distribution.

### D.5 Comparison on sRGB Dataset

We conduct experiments on real-world raw video (transformed to sRGB) denoising. Specifically, we test our denoiser on the indoor test videos (Scenes 7-11) of CRVD (Yue et al., 2020) in Table 3. We use reference-based evaluation and directly test the models on the indoor test set. For fair comparisons, we here compare with RealBasicVSR* because both methods are not trained on CRVD. Our model outperforms RealBasicVSR* by a large margin under the reference-based metrics.

Table 3: Comparisons of ReViD and RealBasicVSR* on CRVD (indoor).

| Reference-based metric | RealBasicVSR* | ReViD (Ours) |
|---|---|---|
| PSNR | 27.41 | **29.61** |
| SSIM | 0.896 | **0.919** |

Processing an existing raw paired dataset as a paired sRGB dataset is possible as an alternative. However, dealing with real-world videos (not synthesized videos from raw images) is the **ultimate** goal of real video denoising. For most images/videos we encounter in our life and on the internet, we do not have access to their raw versions and neither do we have access to the parameters of different sensors and ISP pipelines. Therefore, for practical applications, the no-reference image quality assessment (IQA) metrics (*e.g.*, NIQE, BRISQUE and PIQE) are important and widely used

in existing real-world super-resolution/denoising methods (Zhang et al., 2022; Wang et al., 2021; Zhang et al., 2021). In addition to the non-reference IQA metric, we also compared the visual quality of different methods for the real-world test videos, as shown in Figure 8 in the paper and Figure 3, which we believe can also demonstrate the effectiveness of the proposed method.

In addition, our synthetic setting is very meaningful in real-world applications. Actually, real image denoising/super-resolution methods (Chan et al., 2022b; Zhang et al., 2022; 2021) which use synthetic degradations have already shown promising results and have attracted more and more attention in the low-level computer vision community. The synthetic degradations aim to cover a wide range of reasonable noises from randomized pipelines. These methods have shown better generalization performance than training on collected datasets with a specific camera. Based on this, we make the **first** attempt to propose a new **video** noise degradations in real video denoising. Extensive experiments verify the superiority of our method on real-world videos.

### D.6 Ablation Study on DCN and Multiscale

We conduct ablation studies on DCN and multiscale in Table 4. Specifically, we train all architectures on DAVIS, and calculate average PSNR over all testing noise levels on the DAVIS test set. Training our model without DCN or multiscale degrades the performance, which demonstrate the effectiveness of DCN and multiscale. In addition, training the model with more scales achieves better performance but with the expense of a larger model size (29.02M). To trade-off the performance and model size, we do not use more scales in our architecture.

Table 4: Ablation study on DCN and multiscale on DAVIS test set.

| Methods | w/o DCN | w/o multiscale | w/ more scales | ReViD (Ours) |
|---|---|---|---|---|
| Average PSNR | 36.47 | 36.52 | 37.48 | 37.45 |

### D.7 Comparison on FLOPs

Model inference time was provided in Table 1 in the paper, which can reflect the efficiency of the models. We compared the FLOPs and PSNR performance of different video denoising methods in Table 5. Here, the FLOPs is measured in TITAN RTX GPU with the spatial resolutions of $256\times256$. Our model achieves the best PSNR performance, although it has more FLOPs than BasicVSR++ due to the multi-scales. Besides, our model outperforms VRT with much fewer FLOPs.

Table 5: Comparison with different methods on FLOPs.

| Methods | BasicVSR++ | VRT | ReViD (Ours) |
|---|---|---|---|
| FLOPs (G) | 42.8 | 721.9 | 172.8 |
| PSNR (db) | 36.24 | 37.03 | 37.45 |

### D.8 Results of Video Deblurring

Our main goal is to propose a new realistic degradation model for effective real video denoising. The proposed degradation model and architecture can indeed be further extended to other real-world video restoration tasks. We extend our model for the video deblurring task, and compare our model with EDVR (Wang et al., 2019), STFAN (Zhou et al., 2019), TSP (Pan et al., 2020) and BasicVSR++ (Chan et al., 2022a). Specifically, we train our model on the GoPro dataset (Nah et al., 2017) and show the results in Table 6. Comparing to other competing methods, our model achieves the best PSNR and SSIM. These results further demonstrate the effectiveness and flexibility of our design.

Table 6: Performance on video deblurring on the GoPro test set.

| Methods | EDVR | STFAN | TSP | BasicVSR++ | ReViD (Ours) |
|---|---|---|---|---|---|
| PSNR/SSIM | 26.83/0.843 | 28.59/0.861 | 31.67/0.928 | 34.01/0.952 | 34.23/0.958 |

### D.9 GENERALIZATION OF REAL VIDEO DENOISING MODEL

To investigate the generalization performance, we compare the PSNR of our method with RealBasicVSR (Chan et al., 2022b) using on REDS4 testing set. Note that these two methods are trained on our noise degradations. Specifically, we use REDS4 (4 testing clips, *i.e.*, 000, 011, 015 and 020) to synthesize Gaussian noise, Poisson noise, Speckle noise, Camera noise, JPEG compression noise and Video compression noise using the same setting as Figures 1 and 10. The levels of Gaussian and Speckle noise are 10, the scale of Poisson is 0.05, the quality scale of JPEG compression noise is 80, and the codec and bitrate of Video compression noise are 'mpeg4' and $1e5$. In Table 7, our method achieves higher PSNR than RealBasicVSR (Chan et al., 2022b). It means that our video denoiser has better generalization performance on other noise.

Table 7: Generalization to different kinds of noise on REDS4.

| Methods | Gaussian noise | Poisson noise | Speckle noise | Camera noise | JPEG comp. noise | Video comp. noise |
|---|---|---|---|---|---|---|
| RealBasicVSR* | 26.57 | 26.63 | 26.15 | 26.92 | 26.19 | 25.13 |
| Ours-real | **28.03** | **28.17** | **28.14** | **28.63** | **28.18** | **26.82** |

### D.10 MORE QUALITATIVE COMPARISON

In Figures 2 and 3, we provide more visual comparisons of different video denoising methods for synthetic Gaussian denoising and general real video denoising. Our proposed denoiser restores better structures and preserves clean edge than previous state-of-the-art video denoising methods, even though the noise level is high. In particular, our model is able to synthesize the side profile in the second line of Figure 2. For real video denoising, our model achieves the best visual quality among different methods. For example, our model can generate feather texture of a bird in the third line of Figure 3. There results demonstrate our degradation model is able to improve the generalization ability.

## E LIMITATIONS AND SOCIETAL IMPACTS

Our method achieves state-of-the-art performance in synthetic Gaussian denoising and practical real video denoising. This paper makes the first attempt to propose noise degradations. Our method can be used in some applications with positive societal impacts. For example, it is able to restore old videos and remove compression noise from video in web. However, there are some limitations in practice. First, it is hard for our model to remove blur artifacts which often occur in videos due to exposure time in different cameras. However, our degradation pipeline mainly contains different kind of noise. Second, it is challenging to remove big spot noise. Third, our denoiser is trained with the GAN loss and it may change the identity of details (e.g., human face) especially when the input is severely degraded.

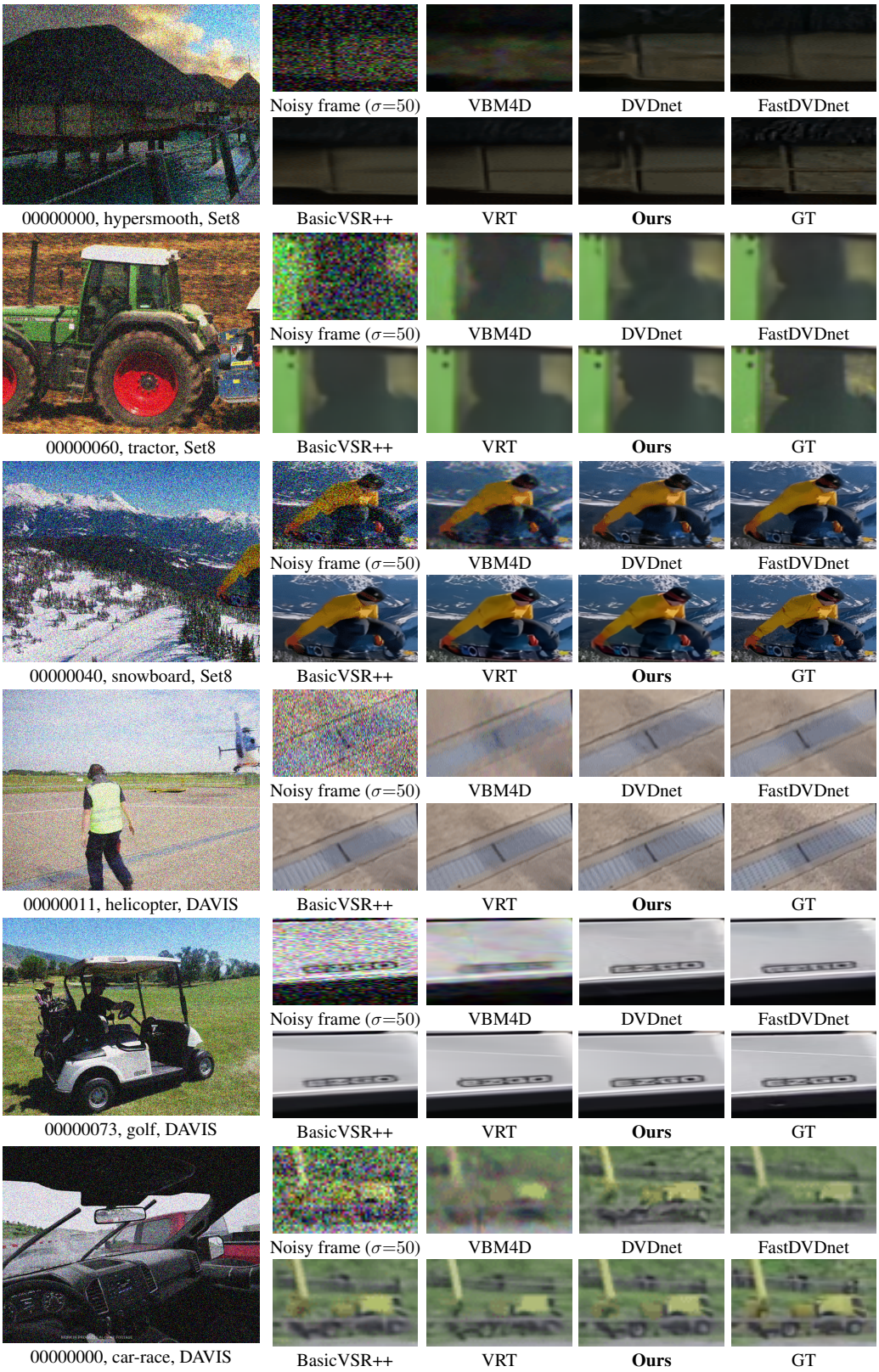


Figure 2: Visual comparison of different methods on DAVIS under the noise level of 50.

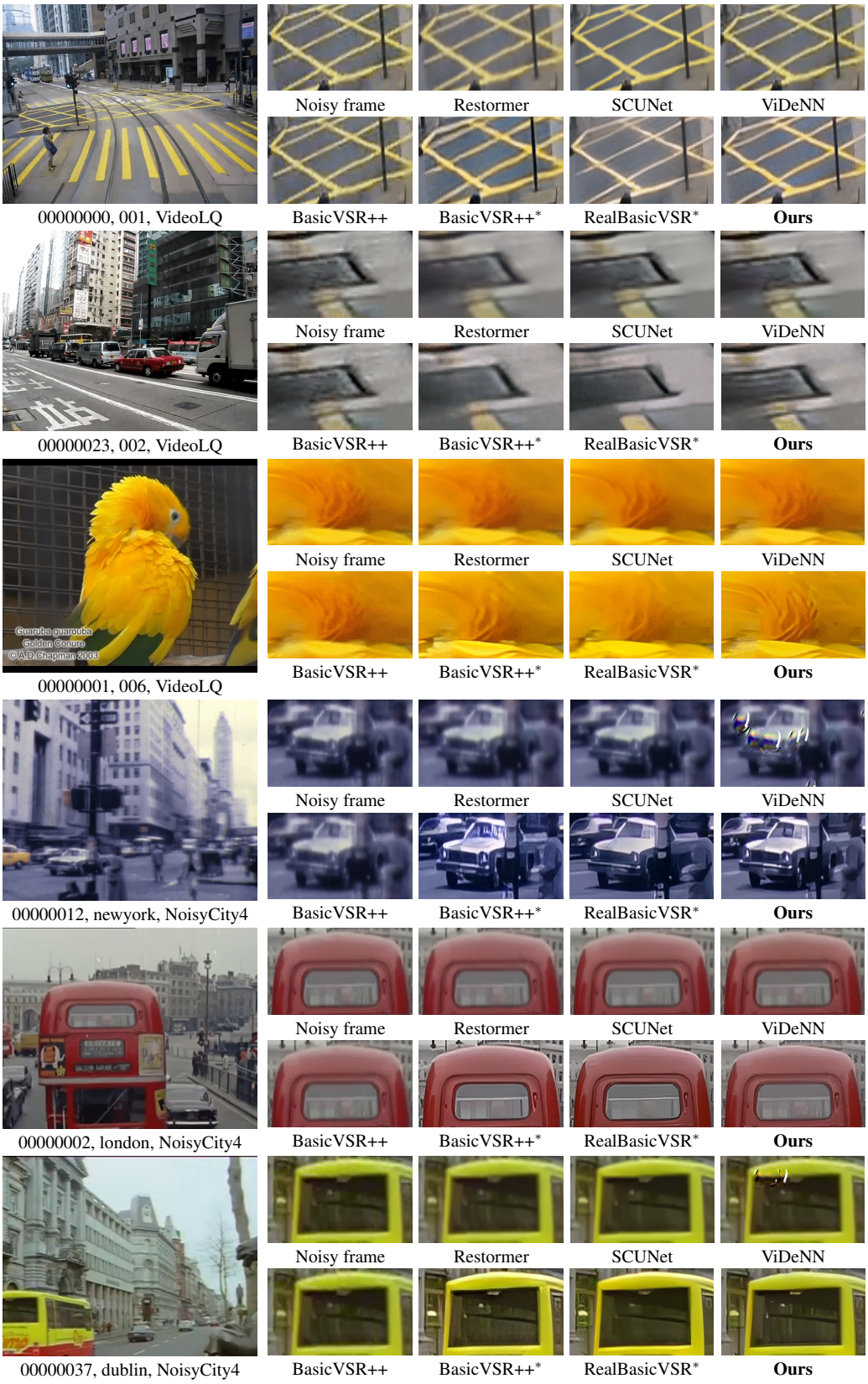


Figure 3: Visual comparison of different video denoising methods on VideoLQ and NoisyCity4.

## REFERENCES

Chris M Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 1995.

Kelvin CK Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022a.

Kelvin CK Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Investigating tradeoffs in real-world video super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022b.

Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *International Conference on Image Processing*, 1994.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition*, 2016.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, 2016.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *IEEE conference on computer vision and pattern recognition*, pp. 3883–3891, 2017.

Jinshan Pan, Haoran Bai, and Jinhui Tang. Cascaded deep video deblurring using temporal sharpness prior. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3043–3051, 2020.

Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *IEEE conference on computer vision and pattern recognition*, 2017.

Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *IEEE International Conference on Computer Vision*, 2021.

Huanjing Yue, Cong Cao, Lei Liao, Ronghe Chu, and Jingyu Yang. Supervised raw video denoising with a benchmark dataset on dynamic scenes. In *IEEE conference on computer vision and pattern recognition*, pp. 2301–2310, 2020.

Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. In *IEEE International Conference on Computer Vision*, 2021.

Kai Zhang, Yawei Li, Jingyun Liang, Jiezhang Cao, Yulun Zhang, Hao Tang, Radu Timofte, and Luc Van Gool. Practical blind denoising via swin-conv-unet and data synthesis. *arXiv preprint arXiv:2203.13278*, 2022.

Shangchen Zhou, Jiawei Zhang, Jinshan Pan, Haozhe Xie, Wangmeng Zuo, and Jimmy Ren. Spatio-temporal filter adaptive network for video deblurring. In *IEEE International Conference on Computer Vision*, pp. 2482–2491, 2019.

Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.