Contrastive Learning for Controllable Blind Video Restoration

Givi Meishvili¹
gmeishvili@microsoft.com
Abdelaziz Djelouah²
abdelaziz.djelouah@disney.com
Sally Hattori³
sally.hattori@disney.com
Christopher Schroers²
christopher.schroers@disney.com

¹ Microsoft
(Work was done at DisneyResearch|Studios, before joining Microsoft)
² DisneyResearch|Studios
Zurich, Switzerland
³ The Walt Disney Company
Los Angles, USA

Abstract

A lot of progress has been made since the first neural network models were trained for specific image restoration tasks, such as super-resolution and denoising. Recently, multi-degradation models have been proposed, allowing for user control of the restoration process needed for real-world applications. However, this aspect is most powerful if the initial restoration can be done as best as possible in a blind setting. In parallel to this line of work, other methods can target the blind setting where, for example, in the case of super-resolution, the blur kernel is estimated for conditioning the restoration part. In particular, discriminative learning has played a key role in pushing the state of the art. Still, the learned representation cannot be interpreted or manipulated and remains a black box that doesn’t offer any possibility for user-guided correction. This work addresses those issues through a representation learning pipeline that helps separate content from degradation by reasoning on pairs of degraded patches. The degradation representation is used as conditioning for a video restoration model that can denoise and upscale to arbitrary resolutions and remove film scratches. Finally, the learned representation can be mutated to fine-tune the restoration results. We demonstrate state-of-the-art results compared to the most recent video super-resolution and denoising methods.

1 Introduction

With the development of video streaming services and the increased competition between the different providers in terms of catalog size, there is a regain of interest for the studios to remaster old shows and productions to make them available on their streaming platform. Our work addresses the problem of video restoration in the context of remastering legacy video content. This content is often available in noisy, blurry, and low-resolution format and may contain scratches. Therefore, the remastering process has to address a combination of degradations and, importantly, allow for user control to have a fine-level control on the output quality. Recent developments in deep learning have pushed the state-of-the-art
in the sub-problems independently by exploring different architectures or training settings in super-resolution \([6, 28, 29, 50, 52, 65]\) and video denoising \([45, 59]\). However, chaining these specialized models is sub-optimal and multi-degradation models have been successfully proposed \([17]\). One essential requirement for adopting a restoration model by the industry is user control of the restoration process. This appears more clearly in recent works such as \([11]\), where image sharpness can be fine-tuned locally. More recently, Kim et al \([25]\) further reduce the complexity of multi-degradation models through architecture search, with the objective of interactive control of the restoration results. This requirement for user control is most powerful if the initial restoration can be done as best as possible in a blind setting, limiting user efforts to minimal fine-tuning. Blind restoration is not possible with these existing models \([17, 25]\) which require providing the degradation parameters. In parallel, most recent blind restoration methods \([49]\) achieve good results. However, the learned representation cannot be interpreted or manipulated.

Here we propose a multi-degradation restoration model that can address jointly denoising, super-resolution and scratch-removal. The restoration can operate in blind settings while still allowing for manipulating the result: The input video can be in low-resolution and may contain scratches. The model automatically estimates the degradation and produces restored frames both in low-resolution and high-resolution. Additionally, the output can be further manipulated to increase sharpness (see Fig. 1). A brief overview of our method during inference is presented in Figure 2. It consists of three main steps: (i) extracting an interpretable and controllable representation of different degradations; (ii) manipulating the degradations if necessary; (iii) finally conditioning the restoration backbone with estimated/manipulated degradation embedding. To the best of our knowledge, no solution considers the complete problem of video restoration that takes into account: Scratch removal, denoising, and up-scaling while offering flexibility in manually fine-tuning the restoration of the signal.

Our training strategy leverages contrastive learning to learn an abstract representation that distinguishes various degradations in the representation space rather than explicit estimation in the pixel space. A key difference from Wang et al. \([49]\) is the possibility of controlling the restoration process via manipulating the degradation features. This requires better estimates for the degradation parameters, which is possible thanks to our training strategy using pairs of degraded training samples and hard negative samples. Finally, we
Figure 2: **Overview of our controllable restoration pipeline.** We first estimate the degradation feature by feeding the corrupted video to the encoder $E_d$. The degradation feature is used as conditioning for the restoration backbone $R_B$. It is possible to adjust both the denoising strength and blur kernel: This mutated version of the embedding (in orange) can similarly be used as conditioning for the restoration. It corresponds to the alternative outputs indicated by the dotted arrows (see text for details).

consider a wider range of degradations and address video restoration in a general setting where super-resolution is not limited to a discrete set of scaling factors is necessary when processing video formats like NTSC. Our contributions can be summarized as follows: (i) A video restoration model that can jointly address multiple types of degradations. (ii) A new contrastive training strategy to learn an interpretable and controllable representation of different degradations. (iii) State-of-the-art results in blind video restoration.

## 2 Related Work

**Super-Resolution.** This research area is active and has primarily benefited from the latest advances in deep learning (see, e.g., [13, 26, 28]). An important part of super-resolution research works has focused on improving task-specific CNN architectures and components (see e.g., [1, 12, 27, 29, 30, 33, 38, 43, 50, 55, 57, 60, 64, 65, 66]). Many other aspects have been considered, ranging from using adversarial training for realistic detail hallucination [4, 28, 37, 62], to improve the realism of the training set through accurate modeling [7, 54] or through using real zoomed-in images [9, 63]. Temporal information can also be used in the context of video super-resolution [6, 16, 22, 24, 31, 44, 47, 51, 56].

However, in this work, we focus more on methods addressing blind super-resolution [2, 11, 15, 34, 40]. These rely on some form of *test-time* optimization to estimate the blur kernel and predict the corresponding high-resolution output. These two steps can be done separately [36], jointly [11, 34] or require a fine-tuning of the super-resolution model [2, 40]. In the case of blind video super-resolution, Pan *et al.* [36] estimate a blur kernel used in an image deconvolution step. The resulting image is then restored using a neural network and aligned adjacent frames. We can note that this strategy may not be optimal as the restoration neural network cannot directly leverage the blur kernel information.

**Denoising.** Similarly to super-resolution, a lot of progress has been made since early works based on neural networks [5, 21, 53]. We focus here on recent video denoising methods:
Yue et al. [59] proposed a raw video denoising network (RViDeNet) by exploring the temporal, spatial, and channel correlations of video frames. Tassano et al. [46] proposed a video denoising algorithm based on a convolutional neural network model conditioned on the noise level. Maggioni et al. [32] introduced a multi-stage algorithm to reduce the complexity while maintaining denoising performance. These methods strongly rely on providing noise level as input. Closer to our blind setting, Claus et al. [10] use a multi-frame neural network architecture to denoise videos and considered varied noise models during training. Although more robust than specialized denoisers, results are not competitive with more recent methods leveraging noise parameters at test time.

Scratch Removal. Scratch removal is a classical mixed degradation problem when working with old photo/video data, and most existing methods consider it an image inpainting problem [3, 8, 14, 41]. Some works consider joint restoration of images corrupted by a combination of different distortions [42, 58]. Wan et al. [48] proposed a triplet domain translation network by leveraging real photos and synthetic image pairs and trained two variational autoencoders (VAEs) to transform old photos and clean photos into two latent spaces. And the translation between these two latent spaces is learned with synthetic paired data.

Multi-degradation models. Combining multiple specialized models to restore images is not optimal and efficient conditioning for a multi-degradation model was proposed by Heet al. [17]. This model was recently used by Wang et al. [49] in the blind restoration setting, leveraging contrastive learning to avoid test-time optimization while still conditioning the restoration model on the estimated degradation. However, the proposed model is limited to images, fixed scaling factors and the learned representation cannot be interpreted or manipulated. Finally, Kim et al. [25] further reduce the complexity of multi-degradation models through architecture search, with the objective of interactive control of the restoration results. They still require providing the degradation parameters.

3 Method

We aim to build a model that can restore videos corrupted by the most common set of degradation present in legacy film content, namely: scratches, noise, and the implicit blur in the low-resolution input. We can briefly formulate the degradation model of a set of consecutive low-resolution (LR) frames $y$ as follows:

$$y = S \circ \left( (x \ast k) \downarrow_s + n \right)$$

where $x$ is the corresponding unknown set of consecutive high-resolution (HR) frames, $\ast$ is convolution operation, $k$ is a blur kernel, $\downarrow_s$ denotes downsampling operation by factor $s$, $n$ stands for noise, and $S$ represents a film scratch as a mask that sets pixel color values to 1.

As illustrated in Figure 2, we train an encoder $E_d$ capable of extracting a latent representation for the degradation present in the input set of frames. For this, we leverage recent advances in contrastive learning [18, 49]. This latent representation is then used as conditioning for the feature restoration backbone $R_F$, which is used both for low-resolution denoising, with $R_{DN}$, and the super-resolution path $R_{SR}$. We leverage information from multiple frames in our model without using explicit motion estimation - a strategy already successfully used in frame interpolation [23]. We use a set of 5 input frames for each output frame: the current frame and 2 temporally adjacent frames from the past and future. Our model also learns to decode the degradation representation into blur kernel and noise levels. Furthermore, it is possible to modify these parameters and adjust the latent representation accordingly, thanks
Figure 3: Overview of our degradation learning pipeline. We first degrade two high-resolution input images with a pair of degradations $d_1$, $d_2$. We encode low-resolution degraded image pairs using encoder $E_d$. Later, features of the first and second rows are concatenated and passed to a two-layer MLP network. Final outputs connected with a green arrow form a positive pair for contrastive learning. A red feature from the third row creates a hard negative example for the feature from the first row since its obtained via encoding the same image corrupted with degradations $d_3$ and $d_4$. We additionally regress the blur kernel $k_1$ and noise level $\sigma_1$ via encoders $E_k$ and $E_s$, respectively. We also learn to manipulate features using encoder $M$ by supplying it with adjusted degradation parameters $k_5, \sigma_5$ and obtain $z_p^5 = M(z_p^6, k_5, \sigma_5)$.

to the mutator model $M$. This flexibility is needed in real-world applications where artists may want to control sharpness levels and denoising strength.

In the following, we first learn video degradation representation, then our proposal to allow the manipulation of the learned latent representation and finally, we take advantage of the learned representation to condition the restoration task.

**Video Degradation Representation** The objective is to learn to extract from the input frames a latent representation that should be discriminative towards different degradations in the input. More precisely, two different, similarly degraded videos should lead to two embeddings close to each other. In contrast, the two differently degraded versions of the same video should result in latent representations further apart. This is a more challenging objective than the one considered by Wang et al. [49], which is a more straightforward application of the Moco [18] representation learning framework: the loss was designed such as to push further away the embedding of patches from different images while bringing closer patches from the same image. Such an objective doesn’t encourage a clear disentanglement between the content and the degradation.

We are interested in disentangling the degradation from the content, but different samples from the training set are captured with sensors of varying resolutions, exposures, and noise levels. Any high-resolution image already contains a certain amount of degradation, and the application of the degradation model from Equation 1 will result in a mixture of two degra-
dations: inherent from a high-resolution image and one from equation 1. Separating these two degradations is an ill-posed problem. Therefore, directly training the encoder $E_d$ with a Multilayer Perceptron (MLP) that tries to optimize our contrastive learning objective is not optimal. To address this issue, we propose to train the encoder $E_d$ using pairs of degraded patches obtained from sampling a random high-resolution image and degrading it with two different degradations. Consequently, the MLP should focus on differences between degradations introduced during training rather than the ones present in the original high-resolution video.

An overview of the training procedure is presented in Figure 3. Let us denote a specific set of different degradations from equation 1 as $d_i \sim \mathcal{D}$ parameterized by blur kernel $k_i$ and noise level $\sigma_i$, $y^i_p = d_i(x_p)$ as video $x_p$ degraded with degradation $d_i$, and $z^i_p = E_d(y^i_p)$ as latent vector obtained by encoding $y^i_p$ using encoder $E_d$. We sample pairs of degradations $(d_i, d_j)$, $(d_k, d_l)$, and videos $x_p, x_q$. We apply pairs of sampled degradations to the videos and encode them using encoder $E_d$: $x_p \rightarrow (d_i(x_p), d_j(x_p)) \rightarrow (y^i_p, y^j_p) \rightarrow (z^i_p, z^j_p)$ $x_q \rightarrow (d_i(x_q), d_j(x_q)) \rightarrow (y^i_q, y^j_q) \rightarrow (z^i_q, z^j_q)$ $x_p \rightarrow (d_k(x_p), d_l(x_p)) \rightarrow (y^k_p, y^l_p) \rightarrow (z^k_p, z^l_p)$

where superscripts and subscripts denote degradations and input videos, respectively. Note that embedding pairs $(z^i_p, z^i_q)$ and $(z^j_q, z^j_q)$ are obtained by degrading two different videos $x_p$ and $x_q$, with the same pair of degradations $(d_i, d_j)$. Therefore, they form a positive pair. Hard negative pairs $(z^k_p, z^l_p)$ and $(z^k_q, z^l_q)$ are obtained by degrading the same video $x_p$ with different pairs of degradations: $(d_i, d_j)$ and $(d_k, d_l)$. We provide these difficult negative examples during training to force the neural representation to focus on the degradation rather than the content. Next, we define the relative degradations via concatenating the resulting embedding pairs of degradations:

$$\psi^i_p = F([z^i_p, z^i_q]), \quad \psi^j_q = F([z^j_q, z^j_q]), \quad \text{and} \quad \psi^{kl} = F([z^k_p, z^l_q]).$$

We want $\psi^i_p$ to be similar to $\psi^j_q$ since they share the same relative degradations and are dissimilar to $\psi^{kl}$ since degradations are different. Therefore, an InfoNCE loss is used to measure the similarity:

$$\mathcal{L}_c = \sum_{p,q} \sum_{i,j} - \log \frac{e(\psi^i_p \cdot \psi^i_q / \tau)}{\sum_{t=1}^{N_Q} e(\psi^i_p \cdot \psi^t / \tau) + e(\psi^{kl} \cdot \psi^i_q / \tau)}$$

where a different degradation pair $kl$ is randomly sampled for each degradation pair $ij$. $N_Q$ is the number of samples in the MoCo queue, $\mathcal{V}$ is a set of training videos, $\mathcal{D}$ is a set of degradations, $\tau$ is a temperature parameter, and $\cdot$ denotes the dot product between two vectors.

To allow the modification of the results and fine-tuning of the outputs in addition to optimizing for $\mathcal{L}_c$ we also estimate the parameters $k_i$ and $\sigma_i$ of applied degradation $d_i$. We train a small degradation regressor MLPs: $E_k$ and $E_s$ that regress the parameters $k_i$ and $\sigma_i$, in a standardized format by optimizing:

$$\mathcal{L}_k = \sum_p \sum_i \left| E_k \left( E_d \left( d_i(x_p) \right) \right) - k_i \right| \quad \mathcal{L}_s = \sum_p \sum_i \left| E_s \left( E_d \left( d_i(x_p) \right) \right) - \sigma_i \right|$$

where subscripts $k$ and $\sigma$ identify the specific output of the model $E$.

Overall training objective can be summarized as follows:

$$\mathcal{L} = \lambda_c \mathcal{L}_c + \lambda_k \mathcal{L}_k + \lambda_s \mathcal{L}_s$$

**Learning to Manipulate Degradations.** Our goal is to restore the distorted videos. However, we also want to have fine-grained control over this process. For example, one might...
need to correct the blur kernel, adjust the noise level and obtain the alternatively restored video. Therefore, we freeze the pre-trained encoder $E_d$ and train the model $M$ to perform manipulations in the latent space of degradations. Given the embedding $z^i_p = E_d(d_i(x_p))$ and some new adjusted parameters $k_j, \sigma_j$, the model $M$ enables the manipulations in the latent space and regresses the feature $z^i_p = M(z^i_p, k_j, \sigma_j)$. During training we sample video $x_p$, and a pair of degradations: $d_i, d_j$. Next, we degrade $x_p$ obtaining $y^i_p = d_i(x_p)$ and $y^j_p = d_j(x_p)$. We compute encodings $z^i_p = E_d(y^i_p), z^j_p = E_d(y^j_p)$ using frozen encoder $E_d$. Finally, we train model $M$ by minimizing the following objective:

$$L_m = \sum_{p} \sum_{i,j} M(z^i_p, k_j, \sigma_j) - z^j_p$$

(5)

**Learning Conditional Restoration.** As illustrated in Figure 2, the proposed model extracts from consecutive frames an encoding of the degradation present in the video. This degradation, expressed as a latent vector, is then used as conditioning for the restoration. Formally our model consists of restoration backbone $R_B$ and two task-specific branches: $R_{SR}$ and $R_{DN}$ for super-resolution and denoising, respectively. The motivation for having a shared backbone $R_B$ is to simultaneously learn features beneficial for different restoration tasks. While the networks $R_{SR}$ and $R_{DN}$ should learn features tailored for super-resolution, denoising, and scratch removal, respectively.

Given a corrupted input $y^i_p$, we first obtain the corresponding degradation embedding $E_d(y^i_p)$. We pass both $y^i_p$ and $E_d(y^i_p)$ to the restoration backbone $R_B$. Consequently, the resulting final feature map from $R_B$ is fed to $R_{SR}$ and $R_{DN}$ subnetworks, respectively. Therefore, we produce two outputs in this model. The first is the low-resolution denoised image and, consequently, the original low-resolution noise. The second is the denoised high-resolution image. Rather than outputting a fixed $4 \times 4$ super-resolved frame, we employ Meta Upscale module [13] at the end of our $R_{SR}$ model to enable non-integer upsampling factors and address more general scenarios. Additionally, our model must remove the possible scratches presented in the video for both super-resolution and denoising branches. Hence in addition to the losses mentioned in the equation 4, during training models $R_{SR}$ and $R_{DN}$ are trained to minimize objectives $L_{SR}$ and $L_{DN}$ respectively.

$$L_{SR} = \sum_{p} \sum_{i} R_{SR}(E_d(y^i_p), \hat{x}_p) - \hat{x}_p \quad L_{DN} = \sum_{p} \sum_{i} R_{DN}(E_d(y^i_p), y^i_p) - (\hat{x}_p * k_i) \downarrow_s$$

(6)

where $\hat{x}_p$ corresponds to the middle high-resolution ground-truth frame of the set of frames $x_p$. In addition to the content losses mentioned in equations 6, we also keep fine-tuning the degradation encoder and manipulation models. Therefore, our final objective becomes:

$$L = \lambda_{SR}L_{SR} + \lambda_{DN}L_{DN} + \lambda_cL_c + \lambda_kL_k + \lambda_sL_s$$

(7)

**4 Experiments**

We incorporated the Vid4 and Set8 datasets for comparison and ablation purposes. We generated multiple degraded versions of the original datasets to demonstrate the capabilities of our pipeline in different settings. First, we created multiple blurry versions of each dataset using nine blur kernels presented in Table 2. Afterward, we downsampled and corrupted each blurry dataset using AWGN of different magnitudes. And finally, we followed the Wan
CONTRASTIVE LEARNING FOR CONTROLLABLE BLIND VIDEO RESTORATION:

<table>
<thead>
<tr>
<th>Feature Contrasting</th>
<th>MAE↓</th>
<th>Kernel Similarity↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single</td>
<td>0.0008</td>
<td>0.9438</td>
</tr>
<tr>
<td>KernelGAN[2]</td>
<td>0.0006</td>
<td>0.9446</td>
</tr>
<tr>
<td>Pairwise</td>
<td>0.0005</td>
<td>0.9821</td>
</tr>
</tbody>
</table>

Table 1: Kernel estimation accuracy for single and pairwise feature contrasting strategies. The first and second rows correspond to single and pairwise feature contrasting strategies, respectively. Results obtained using the KernelGAN[2] are reported in the second row. We report Mean Absolute Error, and Kernel Similarity [20].

et al. [48] method to generate scratched versions of the datasets. For quantitative comparison, we used PSNR and SSIM. Additional training and implementation details can be found in the supplementary material.

Single vs Pairwise Contrasting Ablation. An essential pipeline component is the encoder $E_d$ that learns to map the degraded videos to the latent space. As we mentioned previously, features from the latent space should reflect as much information as possible about the degradation in the input video. Therefore, we evaluate the latent space via the quality of the blur kernels that the encoder $E_k$ produces given a feature from $E_d$ as input. We use Kernel Similarity [20] and Mean Absolute Error (MAE) as evaluation metrics between ground truth and estimated kernels. We thus perform ablation experiments for different ways to train the encoder $E_d$ and justify the choice of the pairwise training strategy. We consider two possible design choices: (i) training $E_d$ by contrasting single video embeddings, and (ii) training $E_d$ by contrasting pairs of video embeddings. MAE’s and Kernel Similarities are reported in Table 1. One can observe that the Pairwise feature contrasting strategy leads to a better quality of the estimated kernels.

Initial vs Mutated Kernels Ablation. We also evaluated how well mutator $M$ manipulates the latent input features. Specifically, we are interested in consistency between the input kernel to the model $M$ and kernel-related information contained in the output manipulated latent code. Towards this goal, we first feed model $M$ with a latent code, noise level, and adjusted blur kernel; and obtain the adjusted code. After, we provide the modified latent code to the Kernel estimator $E_k$ and estimate the adjusted kernel. Finally, we measure the MAE and Kernel Similarity between the initial adjusted blur kernel and the estimated blur kernel after manipulation. We performed the mentioned procedure on degraded videos from Set8 and obtained $\text{MAE} = 0.0004$ and $\text{KS} = 0.9837$ (Kernel Similarity).

Video Super-Resolution Comparisons. We performed a quantitative comparison with the non-blind video super-resolution approach of Tian et al. [47], and with the blind methods of Pan et al. [66], and Zhang et al. [61]. We report results for different blur kernels and noise levels in Table 2. It provides mean PSNR/SSIM per kernel and per-noise level for different methods. This allows analyzing all cases and observing how different methods perform on isotropic/anisotropic gaussian blur kernels and noise levels of different magnitudes. Our method achieves the best performance in all settings except for the one closest to the bicubic kernel, which is the one where naturally a specialized model [47] performs best. To understand the benefits of our multi-frame and pairwise training, we retrained the model of Wang et al. [49] on the Vimeo90K[56] and evaluated it on the Vid4 and Set8 test sets. Retrained model achieves 22.47 / 0.63 and, 26.80 / 0.74 for Vid4 and Set8, respectively. In contrast, our model achieves 22.73 / 0.66 on Vid4 and 27.07 / 0.76 on Set8 while simultaneously addressing multiple restoration tasks, handling non-integer scaling factors, and manipulating results.
### CONTRASTIVE LEARNING FOR CONTROLLABLE BLIND VIDEO RESTORATION:

Table 2: Quantitative comparison to other video super-resolution methods at 4x scaling factor. We report PSNR/SSIM values of our and competitor methods on the Set8 and Vid4 datasets. Different rows and columns correspond to different AWGN levels and blur kernels.

<table>
<thead>
<tr>
<th>σ Method</th>
<th>Blur Kernels</th>
<th>σ Method</th>
<th>Blur Kernels</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ours</strong></td>
<td><strong>All</strong></td>
<td><strong>Ours</strong></td>
<td><strong>All</strong></td>
</tr>
<tr>
<td><strong>Vid4</strong></td>
<td><strong>SET8</strong></td>
<td><strong>Vid4</strong></td>
<td><strong>SET8</strong></td>
</tr>
<tr>
<td><strong>σ</strong> Method</td>
<td><strong>Dataset</strong></td>
<td><strong>σ</strong> Method</td>
<td><strong>Dataset</strong></td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td><strong>40.75 / 0.99</strong></td>
<td><strong>40.22 / 0.99</strong></td>
<td><strong>40.75 / 0.99</strong></td>
</tr>
<tr>
<td><strong>UDV</strong> [39]</td>
<td>36.69 / 0.98</td>
<td>38.11 / 0.97</td>
<td>36.69 / 0.98</td>
</tr>
<tr>
<td><strong>DvDNet</strong></td>
<td>37.92 / 0.99</td>
<td>39.30 / 0.98</td>
<td>37.92 / 0.99</td>
</tr>
<tr>
<td><strong>FastDvDNet</strong></td>
<td>40.68 / 0.99</td>
<td>38.90 / 0.99</td>
<td>40.68 / 0.99</td>
</tr>
<tr>
<td><strong>Dataset</strong></td>
<td><strong>σ</strong> Method</td>
<td><strong>Dataset</strong></td>
<td><strong>σ</strong> Method</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td><strong>35.46 / 0.99</strong></td>
<td><strong>35.09 / 0.99</strong></td>
<td><strong>Ours</strong></td>
</tr>
<tr>
<td><strong>All</strong></td>
<td><strong>33.41 / 0.97</strong></td>
<td><strong>33.96 / 0.95</strong></td>
<td><strong>UDV</strong> [39]</td>
</tr>
<tr>
<td><strong>DvDNet</strong></td>
<td><strong>34.27 / 0.98</strong></td>
<td><strong>34.02 / 0.96</strong></td>
<td><strong>FastDvDNet</strong></td>
</tr>
<tr>
<td><strong>Dataset</strong></td>
<td><strong>σ</strong> Method</td>
<td><strong>Dataset</strong></td>
<td><strong>σ</strong> Method</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td><strong>35.46 / 0.99</strong></td>
<td><strong>35.09 / 0.99</strong></td>
<td><strong>Ours</strong></td>
</tr>
<tr>
<td><strong>All</strong></td>
<td><strong>33.41 / 0.97</strong></td>
<td><strong>33.96 / 0.95</strong></td>
<td><strong>UDV</strong> [39]</td>
</tr>
<tr>
<td><strong>DvDNet</strong></td>
<td><strong>34.27 / 0.98</strong></td>
<td><strong>34.02 / 0.96</strong></td>
<td><strong>FastDvDNet</strong></td>
</tr>
</tbody>
</table>

Table 3: Quantitative comparison to the non-blind video denoising methods. We report PSNR/SSIM values on VID4 and Set8 datasets.

| Ours | 36.09 / 0.99 | 31.93 / 0.98 | Wan et al. [47] | 24.54 / 0.93 | 26.98 / 0.86 |

Table 4: Quantitative comparison to the scratch removal method of Wan et al. [47].

Video Denoising Comparisons. We performed a quantitative comparison with the video denoising methods of Tassano et al. [38, 39], and Sheth et al. [48]. We report results for...
different noise levels in Table 3. Our blind method achieves competitive performance and slightly outperforms the model of [46], which has access to the noise level as input.

![Figure 4: Qualitative Comparison Super-Resolution.](image)

**Video Scratch Removal Comparisons.** We performed a quantitative comparison with the method of Wan et al. [48]. In this experiment, we generated corrupted versions of Vid4 and Set8 by first adding AWGN with $\sigma = 5$ and applying synthetic scratches following Wan et al. [48]’s protocol. We report PSNR/SSIM metrics in Table 4. Our method outperforms the competitor’s method. Note that our pipeline takes scratched videos as input while [48] takes a single scratched frame. A significant performance gap can be explained by our method leveraging information from the temporal dimension, which is not available in the case of [48]. On the other hand, [48] takes the mask of the scratched region as input, which simplifies the restoration process.

**Manipulating Real Videos.** A real video restoration example is presented in Figure 1. One can observe the gradual decrease of the blur level in restored frames from left to the right. Initially, we pass the feature from encoder $E_d$ to backbone $R_b$ and obtain the results in the 4-th column. We manipulate the blur kernel to both more and less blur. We feed the mutator $M$ with the modified blur kernels to obtain new embeddings to condition the restoration. One can see the effect from blurry to sharper results.

5 Conclusion

In this paper, we proposed a discriminative learning strategy that helps separate content from degradation by reasoning on pairs of degraded patches, where both content and degradation vary independently. The degradation representation is used as conditioning for a video restoration model that can handle denoising, super-resolution, and scratch removal. More importantly, the learned representation can be manipulated to fine-tune the results, which is crucial for real application scenarios.
References


CONTRASTIVE LEARNING FOR CONTROLLABLE BLIND VIDEO RESTORATION:


