Unsupervised Flow-Aligned
Sequence-to-Sequence Learning for Video Restoration

Jing Lin [* 1] **Xiaowan Hu** [* 1] **Yuanhao Cai** [1] **Haoqian Wang** [† 1]
Youliang Yan [2] **Xueyi Zou** [† 2] **Yulun Zhang** [3] **Luc Van Gool** [3]

Abstract

How to properly model the inter-frame relation within the video sequence is an important but unsolved challenge for video restoration (VR). In this work, we propose an unsupervised flow-aligned sequence-to-sequence model (S2SVR) to address this problem. On the one hand, the sequence-to-sequence model, which has proven capable of sequence modeling in the field of natural language processing, is explored for the first time in VR. Optimized serialization modeling shows potential in capturing long-range dependencies among frames. On the other hand, we equip the sequence-to-sequence model with an unsupervised optical flow estimator to maximize its potential. The flow estimator is trained with our proposed unsupervised distillation loss, which can alleviate the data discrepancy and inaccurate degraded optical flow issues of previous flow-based methods. With reliable optical flow, we can establish accurate correspondence among multiple frames, narrowing the domain difference between 1D language and 2D misaligned frames and improving the potential of the sequence-to-sequence model. S2SVR shows superior performance in multiple VR tasks, including video deblurring, video super-resolution, and compressed video quality enhancement. https://github.com/linjing7/VR-Baseline

1. Introduction

Video restoration (VR) aims to reconstruct high-quality (HQ) video from its degraded low-quality (LQ) counterpart, including video deblurring (Xiang et al., 2020), video super-

---

[*]Equal contribution [1]Shenzhen International Graduate School, Tsinghua University [2]Huawei Noah's Ark Lab [3]ETH Zürich. Correspondence to: Xueyi Zou <zouxueyi@huawei.com>, Haoqian Wang <wanghaoqian@tsinghua.edu.cn>.
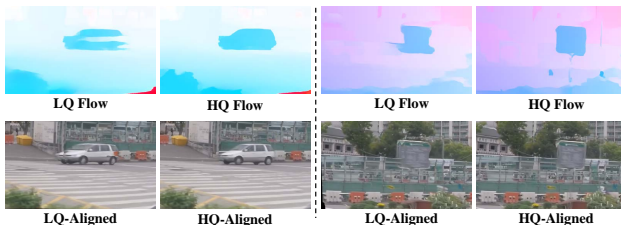
Figure 1. Optical flow estimated from LQ and HQ videos respectively (top), and visual comparison of the aligned frames (bottom).

resolution (SR) (Sajjadi et al., 2018; Chan et al., 2021), and compressed video enhancement (Guan et al., 2019).

Task-driven networks often have complex structures that are elaborately designed for a specific task. These methods may be inapplicable when transferred to a new scenario or a different video restoration task (Yang et al., 2018b; Deng et al., 2020; Cao et al., 2021). Therefore, it is of great significance to explore a unified and versatile framework that can be used for multiple video restoration tasks.

Early works simply extend single image restoration (Dai et al., 2015; Shahar et al., 2011; Liao et al., 2015; Cai et al., 2021) to video restoration. These image-based methods ignore inter-frame correlation, leading to limited performance. Some CNN-based methods (Wang et al., 2019; Pan et al., 2020; Deng et al., 2020) utilize information from the frames within a short temporal window. The ignorance of distant frames significantly limits the performance of these methods. Some researchers use the recurrent neural network (RNN) (Isobe et al., 2020; Yang et al., 2019; Zhong et al., 2020; Chan et al., 2021) to propagate the hidden state in the time domain to expand the temporal receptive field. However, as analyzed in (Jozefowicz et al., 2015), RNN suffers from both exploding and vanishing gradients. As a result, RNN is difficult to learn the long-term dependencies and can not be stacked into very deep models, limiting the representation capacity of restoration network. The transformer-based model (Cao et al., 2021; Lin et al., 2022a) can process a video sequence in parallel with self-attention mechanism. Nonetheless, the model complexity is quadratic to the number of tokens. For video restoration with an immense number of tokens, modeling long-range dependencies means huge computational costs and memory occupation. Thus,

the problem of modeling long-term inter-frame relations with an affordable cost remains formidable.

Based on the sequence nature of videos, our insight into this problem is to treat it as a sequence modeling task and try to solve it with the sequence-to-sequence (seq2seq) model. Seq2seq model has proven capable of sequence modeling (Sutskever et al., 2014; Chopra et al., 2016; Ott et al., 2018; Chen et al., 2018) in the field of natural language processing (NLP), showing great potential in modeling the inter-frame relation within the video sequence. Seq2seq model is devised to serially encode the input sequence into latent vectors and then dynamically decode a target sequence out of that representations. However, the migration of the seq2seq model is inevitably hindered by the domain discrepancy between NLP and VR. The video signal is composed of multiple misaligned 2D frames, while the seq2seq model can only handle continuous 1D input (*e.g.,* , language sequence, time series) canonically. So we need to establish accurate correspondences among multiple frames by performing a spatial alignment with optical flow estimator.

Previous flow-based (Wedel et al., 2009; Sun et al., 2018; Teed & Deng, 2020) methods perform spatial alignment with a pretrained optical flow network. (Chan et al., 2021) prove that feature alignment, *i.e.*, estimating optical flow from the LQ videos and using it to warp the hidden state, can yield a better restoration result than image alignment. However, these flow-based methods may be suboptimal and suffer from the following issues: Firstly, the data discrepancy between synthetic flow dataset and real-world video affects the performance of the pretrained optical flow module in VR. Secondly, the optical flow estimated from the LQ input video (LQ flow) may be unreliable since the video degradation may seriously distort video contents and break pixel-wise correspondences between frames (Zheng et al., 2021). As shown in Fig. 1, the LQ flows lose some motion details, and the frames aligned by the LQ flows (LQ-aligned frames) contain blurry edges. In contrast, the HQ flow is more detailed, and the HQ-aligned frames contain sharper semantics. Besides, for feature alignment, the motion information estimated from the LQ video may be inconsistent with that of the hidden state, which is expected to be spatially aligned with the HQ video. So some artifacts will be brought when the LQ flow is used for feature alignment.

We attempt to address the data discrepancy and inaccurate LQ flow issues with unsupervised distillation optical flow loss. To be specific, we train an optical flow estimator on the VR dataset with unsupervised loss. The data discrepancy naturally disappear since the training and testing dataset both come from the real-world VR dataset. Furthermore, a novel data distillation loss is designed to generate more accurate LQ flows, in which the optical flows estimated from the HQ video serve as the pseudo-labels of the LQ

flows. This loss encourages the LQ flows to imitate the HQ flows, which are more accurate and spatial consistent with the motion information of the hidden state.

Therefore, the unsupervised flow-aligned sequence-to-sequence model is proposed for video restoration tasks (S2SVR). We migrate and improve the seq2seq model from NLP to VR task, and maximize the potential of the seq2seq model with an unsupervised optical flow estimator. In a nutshell, our contributions can be summarized as follows:

- This is the first VR work to explore the sequence-to-sequence model, which comes from NLP and is intrinsically suitable for video sequence modeling.

- The proposed unsupervised distillation optical flow loss alleviates the data discrepancy and inaccurate LQ flow issues of previous flow-based methods, narrowing the domain difference between NLP and VR.

- Extensive experiments show that our method achieves state-of-the-art performance in three typical video restoration tasks, including video deblurring, video super-resolution, and compressed video enhancement.

2. Related Work

2.1. Video Restoration

Early work (Takeda et al., 2009; Shahar et al., 2011; Dai et al., 2015) adopt an image restoration model for video restoration and do not take advantage of information in the neighbouring frames. The ignorance of the inter-frame correlation severely limits the restoration result. Some CNN-based methods (Deng et al., 2020; Tian et al., 2020) employ deformable convolution to perform feature-level alignment. The RNN-based methods design the recurrent structure and attempt to model the long-term dependencies by propagating the hidden state (Isobe et al., 2020; Yang et al., 2019; Zhong et al., 2020). (Chan et al., 2021) prove that the combination of bidirectional propagation and optical flow estimation can achieve ideal results. (Deng et al., 2021) propose a recurrent model with separable-patch architecture and multi-scale integration scheme for fast and accurate video deblurring. However, the RNN-based methods inevidently suffer from the vanishing gradient problem and have difficulty in capturing the long-range temporal dependencies. Recently, the emerging Transformer model has been applied in image and video restoration tasks (Cai et al., 2022a; Liang et al., 2022; Lin et al., 2022b; Cao et al., 2021; Cai et al., 2022b). Nonetheless, the token-based self-attention module has enormous computational and memory cost in restoring long video sequence. Thus, the problem of effectively modeling long-range temporal dependencies within the video sequence remains formidable.

2.2. Sequence-to-Sequence Learning

Seq2seq model is first proposed by (Sutskever et al., 2014) for the machine translation task in which a long short-term
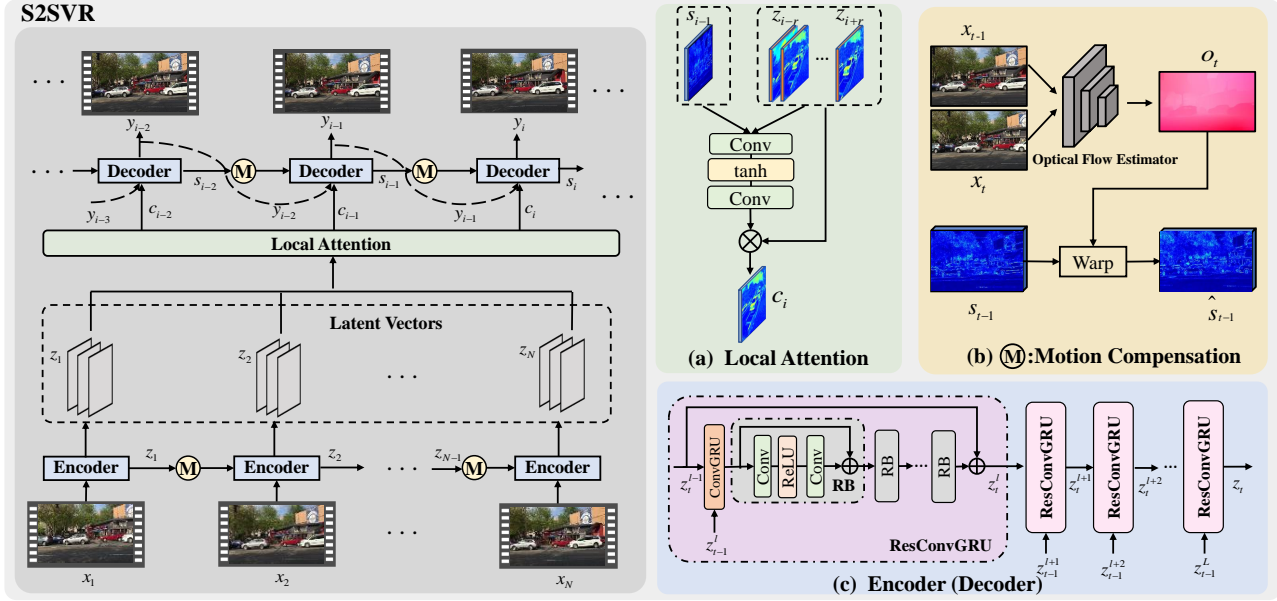
Figure 2. The architecture of the proposed unsupervised flow-aligned seq2seq model (S2SVR). The modules with different background colors on the right show the internal details of (a) local attention, (b) motion compensation and (c) Encoder (Decoder).

memory (LSTM) encodes the input sequence into a latent representation and then another LSTM decodes the target sequence out of that representation. The model is intrinsically suitable for long-range coding tasks. Various variants of the seq2seq model have been applied to many sequence modeling tasks, such as speech recognition (Venugopalan et al., 2015), time series analysis (Kuznetsov & Mariet, 2019; He et al., 2021), and text summarization (Shi et al., 2021). Due to the fundamental difference between video and language, the potential of this serialized encoding-decoding structure in assisting continuous-frame VR is unexplored.

2.3. Optical Flow Estimation

With the development of deep learning, some optical flow estimation networks (Sun et al., 2018; Teed & Deng, 2020) trained on synthetic datasets have achieved better results than non-learning methods (Mémin & Pérez, 1998; Wedel et al., 2009). The domain difference between synthetic optical flow and real-world optical flow datasets leads to limited model performance. (Wang et al., 2018a) suggest using an unsupervised optical flow estimator to circumvent the need for labels. (Wang et al., 2018b) improve the performance of unsupervised optical methods by proposing a new warping module to facilitate large motion learning and model occlusion explicitly. (Shi et al., 2017) train a task-oriented flow module jointly with the video enhancement module in the supervision of $\mathcal{L}_1$ loss. But the jointly-trained flow module becomes unsuitable when cooperating with other video processing modules. Besides, they have not solved the problem that it's difficult to estimate accurate motion information from the severely degraded input frames. Based on the LQ-HQ paired characteristics of VR tasks, we propose a data distillation loss to improve the quality of the LQ flows.

3. Method

In this section, we present our S2SVR model. We first introduce the overall framework of the seq2seq model. Then, we explain the unsupervised distillation optical flow method, which narrow the domain discrepancy between NLP and VR and improve the potential of the seq2seq model in VR.

3.1. Sequence-to-Sequence Learning

To promise that the scalable seq2seq architectures and their efficient implementations can be preserved, S2SVR follows the seq2seq framework from NLP as closely as possible. As shown in Fig. 2, S2SVR is composed of four components: encoder, decoder, local attention, and optical flow estimator.

For notation, we use capital letters to represent sequences,(*e.g.*, X,Y), lower case to denote individual frames in a sequence, (*e.g.*, x_1, x_2). Let $X = \{x_1, x_2, \ldots, x_N\}$ represent the input low-quality video sequence and $Y = \{y_1, y_2, \ldots, y_N\}$ be the corresponding high-quality video sequence, where N is the length of the sequence. The goal of our S2SVR is to estimate the conditional probability of the target sequence respective to the input sequence $P(Y|X)$.

Encoder. Firstly, the encoder read sequentially each $x_i \in X$ and transforms the source sequence into a list of latent vectors $Z = \{z_1, z_2, \ldots, z_N\}$:

$$z_i = F_e(z_{i-1}, x_i), \tag{1}$$

where z_i denotes the latent vector at time step i, and F_e denotes the function of the encoder, which in our implementation is a residual stacked ConvGRU (ResConvGRU). The ResConvGRU will be introduced in the next subsection.

Decoder. Next, the decoder sequentially produces the out-

put video based on the encoded vectors. Specifically, using the chain rule, the conditional probability $P(Y|X)$ can be decomposed as:

$$
\begin{aligned}
P(Y|X) &= P(y_1, y_2, \ldots, y_N | z_1, z_2, \ldots, z_N) \\
&= \prod_{t=1}^{N} P(y_t | y_1, \ldots, y_{t-1}; z_1, \ldots, z_N).
\end{aligned} \quad (2)
$$

We serially generate the subsequent output based on the source sequence encoding and the decoded sequence so far:

$$
y_t = F_d(y_1, y_2, \ldots, y_{t-1}; z_1, z_2, \ldots, z_N). \quad (3)
$$

F_d represents the decoder, which is composed of a ResConvGRU and a feed-forward network. ResConvGRU generates a hidden state s_i, and then s_i passes through the feed-forward network to produce the output frame:

$$
\begin{aligned}
s_i &= F_r(s_{i-1}, y_{i-1}, c_i), \\
y_i &= F_f(s_i),
\end{aligned} \quad (4)
$$

where F_r is the ResConvGRU and F_f denotes the feed-forward network. s_i, y_i refer to the hidden state of ResConvGRU and output frame at i^{th} time step, respectively. And c_i is a context vector generated by the local attention module based on the latent vectors $Z = \{z_1, z_2, \ldots, z_N\}$.

Local Attention. As shown in Fig. 2(a), the attention module generates a context vector c_i for each time step, allowing the decoder to extract information from different parts of the input sequence. Specifically, we represent the context vector c_i as a weighted sum of a subset of the latent vectors:

$$
c_i = \sum_{j=i-r}^{i+r} \alpha_{ij} z_j, \quad (5)
$$

where r is the the subset radius and the weight α_{ij} is:

$$
\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=i-r}^{i+r} exp(e_{ik})}. \quad (6)
$$

$e_{ij} = F_a(s_{i-1}, z_j)$ is an attention model scoring the correspondence between the ith input and the jth output based on s_{i-1} and z_j. Similar to (Shahar et al., 2011), a two-layer feed-forward network is adopted as the attention model:

$$
e_{ij} = \mathbf{V}_a \cdot tanh(\mathbf{W}_a[s_{i-1}, z_j]), \quad (7)
$$

where $\mathbf{V}_a$ and $\mathbf{W}_a$ denote the first and second convolution layers of the feed-forward network, respectively. And $[\cdot, \cdot]$ refers to concatenation along the channel dimension.

Motion Compensation. To improve the performance of the seq2seq model in VR, we need to establish accurate spatial correspondences among multiple frames. Similar to previous methods (Isobe et al., 2020; Yang et al., 2019;
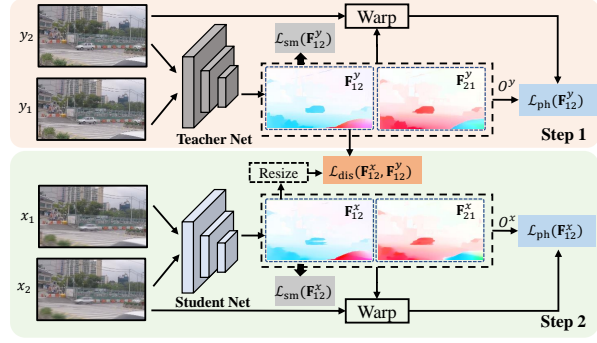

Figure 3. Illustration of the unsupervised optical flow method.

Zhong et al., 2020; Chan et al., 2021), we adopt an optical flow estimator for motion compensation. Specifically, as shown in Fig. 2(b), we employ a flow estimator to predict the motion between two consecutive frames. Then we warp the hidden state of ResConvGRU at last time step s_{t-1}, making it spatially aligned with the input at the current step:

$$
\begin{aligned}
o_t &= F_o(x_t, x_{t-1}), \\
\hat{s}_{t-1} &= F_w(s_{t-1}, o_t),
\end{aligned} \quad (8)
$$

where F_o and F_w respectively refer to the optical flow estimator and spatial warping module. o_t is the optical flow field between the adjacent input frames x_t and x_{t-1}.

3.2. Residual Stacked ConvGRU

We use a deep-stacked ConvGRU for both the encoder and the decoder. Considering the video characteristics, as shown in Fig. 2(c), we make two modifications to the original ConvGRU. Firstly, to improve the image processing ability, several residual blocks are concatenated after the ConvGRU. Besides, motivated by the idea of modeling the difference between an intermediate layer's output and the target, we introduce residual among the layers in a stack. We define the ConvGRU and residual blocks as $F_g(\cdot)$ and $F_b(\cdot)$:

$$
z_t^l = z_t^{l-1} + F_b(F_g(z_{t-1}^l, z_t^{l-1})), \quad (9)
$$

where z_t^l denote the hidden state of lth ConvGRU at time step t. In this way, the vanishing gradient problem can be addressed, allowing us to model the long-term temporal dependencies. More details are provided in the appendix.

3.3. Unsupervised Optical Flow Estimator

As analyzed in Sec. 1, previous flow-based motion compensation methods suffer from the data discrepancy between synthesized and real-world datasets, as well as inaccurate LQ flows. To solve these problems, we propose an unsupervised scheme equipped with a novel distillation loss to train the flow estimator on the VR dataset as shown in Fig. 3.

Let X denotes a LQ input video, and Y is the corresponding HQ video. Our goal is to train a flow network F_o that can estimate accurate motion information from the LQ videos (HQ

videos are unavailable during inference) by predicting the optical flow $\mathbf{F}_{12}^x$ for two consecutive LQ frames $\{x_1, x_2\}$:

$$\mathbf{F}_{12}^x = F_o(x_1, x_2). \qquad (10)$$

The unsupervised scheme is summarized in Algorithm 1 to better understand the proposed unsupervised optical flow estimation method. Firstly, we train a teacher flow estimation network parameterized by θ_t on the HQ videos with photometric loss and smooth loss. After convergence, we use the pretraied teacher estimator to generate pseudo-labels and train a student flow network parameterized by θ_o on the LQ video. In the following, we explain the proposed unsupervised optical flow training scheme step by step.

Step 1. We train an optical flow estimator F_t with photometric loss and smooth loss on the HQ video Y. This optical flow estimator F_t will be frozen and serves as a teacher network in the next step. The photometric loss (Yu et al., 2016) is based on the assumption that the same object in two consecutive frames must have similar intensities:

$$\mathcal{L}_{\text{ph}}(\mathbf{F}_{12}^y) = \sum_p \rho(y_1(p), y_2(p + \mathbf{F}_{12}^y(p))) \cdot O^y(p), \quad (11)$$

where p is the coordinate and O^y is the occlusion mask to discard the loss on the occurred region generated by the bidirectional checking (Wang et al., 2018b), $\rho(\cdot)$ is the ℓ_1 loss, and $\mathbf{F}_{12}^y$ is the optical flow field for two consecutive frames in the HQ videos Y:

$$\mathbf{F}_{12}^y = F_t(y_1, y_2). \qquad (12)$$

Further, we adopt a one-order smooth loss (Godard et al., 2017) to encourage collinearity of neighboring flows:

$$\mathcal{L}_{\text{sm}}(\mathbf{F}_{12}^y) = \sum_{d \in x,y} \sum_p |\partial_d \mathbf{F}_{12}^y(p)| e^{-|\partial_d y_1(p)|} \qquad (13)$$

And then we formulate the loss used in the first step as:

$$\mathcal{L} = \omega_{\text{ph}} \cdot \mathcal{L}_{\text{ph}}(\mathbf{F}_{12}^y) + \omega_{\text{sm}} \cdot \mathcal{L}_{\text{sm}}(\mathbf{F}_{12}^y). \qquad (14)$$

We respectively set the weights ω_{ph} and ω_{sm} to 0.15 and 50.

Step 2. Now we have trained a teacher optical flow estimator F_t which can predict the accurate optical flow $\mathbf{F}_{12}^y$ for two consecutive HQ frames $\{y_1, y_2\} \in Y$:

$$\mathbf{F}_{12}^y = F_t(y_1, y_2). \qquad (15)$$

Based on the assumption that the HQ flow is more accurate for motion compensation, we use $\mathbf{F}_{12}^y$ as the pseudo-labels of the LQ flows F_{12}^x and and propose the distillation loss:

$$\mathcal{L}_{\text{dis}}(\mathbf{F}_{12}^x, \mathbf{F}_{12}^y) = \sum_p |\mathbf{F}_{12}^y(p) - F_u(\mathbf{F}_{12}^x)(p)|, \qquad (16)$$

Algorithm 1 Unsupervised Distillation Optical Flow Loss

Inputs: teacher and student net parameterized by θ_t, θ_o, cost function parameters: loss weights $\{\omega_{\text{ph}}, \omega_{\text{sm}}, \omega_{\text{dis}}\}$, optimization parameters: number of iterations T
Output: pretrained student network
// Step1: train the teacher network
for $j = 0$ to T **do**
    compute photometric loss $\mathcal{L}_{\text{ph}}$ (using Eq. (11))
    compute smooth loss $\mathcal{L}_{\text{sm}}$ (using Eq. (13))
    $\mathcal{L}_{\text{tot}} = \omega_{\text{ph}} \cdot \mathcal{L}_{\text{ph}} + \omega_{\text{sm}} \cdot \mathcal{L}_{\text{sm}}, \quad \nabla L(\theta_t) = \frac{\partial L_{tot}}{\partial \theta_t}$,
    $\theta_t = \theta_t - \alpha \nabla L(\theta_t)$
end for
// Step2: train the student network
for $j = 0$ to T **do**
    compute photometric loss $\mathcal{L}_{\text{ph}}$ (using Eq. (11))
    compute smooth loss $\mathcal{L}_{\text{sm}}$ (using Eq. (13))
    compute data distillation loss $\mathcal{L}_{\text{dis}}$ (using Eq. (16))
    $\mathcal{L}_{\text{tot}} = \omega_{\text{ph}}\mathcal{L}_{\text{ph}} + \omega_{\text{sm}}\mathcal{L}_{\text{sm}} + \omega_{\text{dis}}\mathcal{L}_{\text{dis}}, \; \nabla L(\theta_o) = \frac{\partial \mathcal{L}_{tot}}{\partial \theta_o}$,
    $\theta_o = \theta_o - \alpha \nabla L(\theta_o)$
end for
Return student network parameters θ_o

---

where F_u is a upsample operation to ensure that $\mathbf{F}_{12}^x$ has the same size as $\mathbf{F}_{12}^y$ in video super-resolution task. Along with the photometric loss and smoothness regularization, we train the student flow estimator F_o on the LQ dataset:

$$\mathcal{L} = \omega_{\text{ph}}\mathcal{L}_{\text{ph}}(\mathbf{F}_{12}^x) + \omega_{\text{sm}}\mathcal{L}_{\text{sm}}(\mathbf{F}_{12}^x) + \omega_{\text{dis}}\mathcal{L}_{\text{dis}}(\mathbf{F}_{12}^x, \mathbf{F}_{12}^y). \qquad (17)$$

We set the weights to $\{\omega_{\text{ph}} = 0.15, \omega_{\text{sm}} = 50, \omega_{\text{dis}} = 0.1\}$. The student network will be later used as our optical flow estimator for motion compensation as in Eq. (8). In implementation, we adopt a lightweight flow model pwclite (Liu et al., 2020) as our optical flow network.

4. Experiments

4.1. Implementation Details

Datasets. For video SR, the benchmark datasets consist of REDS4 (Nah et al., 2019a) and Vimeo-90K-T (Xue et al., 2019). For video deblurring, we use the GOPRO dataset (Nah et al., 2017), where 22 videos are used for training and 11 videos for testing. For compressed video enhancement, our models are trained with the MFQEv2 dataset (Guan et al., 2019) including 108 lossless videos. We adopt the dataset from ITU-T (Ohm et al., 2012) containing 18 videos for evaluation. We compress videos by HEVC reference software HM16.5 under Low Delay P (LDP) configuration (Guan et al., 2019; Deng et al., 2020). Evaluation metrics include PSNR and SSIM (Wang et al., 2004).

Settings. Models are trained with nature videos and their degraded counterparts. During unsupervised optical flow training, the learning rate is set to 1×10^{-4}. And during restoration training, the initial learning rate of the flow esti-

Figure 4. Visual comparison of video 4× SR results on the REDS4 (Nah et al., 2019a) dataset. Please zoom in for a better comparison.

Methods	Params	REDS4	Vimeo-90K-T
Bicubic	-	26.14 / 0.7292	31.32 / 0.8684
TOFlow	-	27.98 / 0.7990	33.08 / 0.9054
DUF	5.8 M	28.63 / 0.8251	-
RBPN	12.2 M	30.09 / 0.8590	37.07 / 0.9435
EDVR-M	3.3 M	30.53 / 0.8699	37.09 / 0.9446
EDVR	20.6 M	31.09 / 0.8800	37.61 / 0.9489
PFNL	3.0 M	29.63 / 0.8502	36.14 / 0.9363
MuCAN	-	30.88 / 0.8750	37.32 / 0.9465
BasicVSR	6.3 M	31.42 / 0.8909	37.18 / 0.9450
IconVSR	8.7 M	<u>31.67</u> / <u>0.8948</u>	37.47 / 0.9476
VSR-Transformer	32.6 M	31.19 / 0.8815	**37.71 / 0.9494**
S2SVR (Ours)	13.4 M	**31.96 / 0.8988**	<u>37.63</u> / <u>0.9490</u>

Table 1. Quantitative comparison (PSNR/SSIM) on the video SR dataset REDS4 and Vimeo-90K-T. Bold and underlined text indicate the best and the second-best performance, respectively.

mator and the other modules are set to 5×10^{-5} and 2×10^{-4}, respectively. We use PyTorch to implement our models and train them on 8 Tesla V100 GPUs. More details are provided in the supplementary material due to space limitation.

4.2. Video Super-Resolution

Quantitative Comparison. We compare our method with previous methods: TOFlow (Xue et al., 2019), DUF (Jo et al., 2018), RBPN (Haris et al., 2019), EDVR-M (Wang et al., 2019), EDVR (Wang et al., 2019), PFNL (Yi et al., 2019), MuCAN (Li et al., 2020), BasicVSR (Chan et al., 2021), IconVSR (Chan et al., 2021), and VSR-Transformer (Cao et al., 2021). As shown in Tab. 1, it is clear that our method outperforms all other models by a large margin on the REDS4 dataset. Specifically, our S2SVR model achieves 0.29dB gain over the suboptimal model and 0.77dB over the VSR-Transformer model in PSNR. For Vimeo-90K-T,

Methods	Params	PSNR (dB)	SSIM
Tao *et al.*	-	30.29	0.9014
Su *et al.*	15.30 M	27.31	0.8255
Kim *et al.*	-	26.82	0.8245
Nah *et al.*	-	29.97	0.8947
EDVR	23.6 M	26.83	0.8426
STFAN	5.37 M	28.59	0.8608
TSP	16.19 M	<u>31.67</u>	**0.9279**
UHDVD	-	31.33	0.9210
S2SVR (Ours)	8.44 M	**31.81**	<u>0.9231</u>

Table 2. Video deblurring performance comparison and model parameter analysis on the GOPRO dataset (Nah et al., 2017).

our performance is slightly lower than VSR-Transformer, but S2SVR only requires 41% parameters compared with the latter. It shows that we only need half the parameters to obtain comparable performance to transformer-based models. Note that Vimeo-90K-T contains sequences with seven frames. So it also indicates that our method performs better in restoring long sequences. Serialized modeling of seq2seq models and accurate optical flow estimation facilitates the capture of long-range inter-frame dependencies.

Visual Comparison. From the comparison with other methods in Fig. 4, our S2SVR network has shown great advantages in the restoration of textures and structural details, such as license plate numbers, pane lines, and hairs. Our results are more reliable and detailed, while the other methods suffer from excessive smoothing and content distortion.

4.3. Video Deblurring

Quantitative Comparison. We compare our method against state-of-the-art algorithms, including Tao *et al.* (Tao et al., 2018), Su *et al.* (Su et al., 2017), Kim *et al.*

QP		Approach	AR-CNN (Dong et al., 2015)	DnCNN (Zhang et al., 2017)	DS-CNN (Yang et al., 2018a)	MFQE 1.0 (Yang et al., 2018b)	MFQE 2.0 (Guan et al., 2019)	STDF-R3L (Deng et al., 2020)	S2SVR (Ours)
		Metrics	ΔPSNR / ΔSSIM						
37	A	Traffic	0.239 / 47	0.238 / 57	0.286 / 60	0.497 / 90	0.585 / 102	0.730 / 115	**0.851 / 138**
		PeopleOnStreet	0.346 / 75	0.414 / 82	0.416 / 85	0.802 / 137	0.920 / 157	1.250 / 196	**1.385 / 216**
	B	Kimono	0.219 / 65	0.244 / 75	0.249 / 75	0.495 / 113	0.550 / 118	0.850 / 161	**1.055 / 195**
		ParkScene	0.136 / 38	0.141 / 50	0.153 / 50	0.391 / 103	0.457 / 123	0.590 / 147	**0.649 / 165**
		Cactus	0.190 / 38	0.195 / 48	0.239 / 58	0.439 / 88	0.501 / 100	0.770 / 138	**0.828 / 152**
		BQTerrace	0.195 / 28	0.201 / 38	0.257 / 48	0.270 / 48	0.403 / 67	0.630 / 106	**0.654 / 115**
		BasketballDrive	0.229 / 55	0.251 / 58	0.282 / 65	0.406 / 80	0.465 / 83	0.750 / 123	**0.972 / 157**
	C	RaceHorses	0.219 / 43	0.253 / 65	0.267 / 63	0.340 / 55	0.394 / 80	0.550 / 135	**0.854 / 203**
		BQMall	0.275 / 68	0.281 / 68	0.330 / 80	0.507 / 103	0.618 / 120	0.990 / 180	**1.080 / 205**
		PartyScene	0.107 / 38	0.131 / 48	0.174 / 58	0.217 / 73	0.363 / 118	**0.680** / 194	0.628 / **236**
		BasketballDrill	0.247 / 58	0.331 / 68	0.352 / 68	0.477 / 90	0.579 / 120	0.790 / 149	**0.949 / 179**
	D	RaceHorses	0.268 / 55	0.311 / 73	0.318 / 75	0.507 / 113	0.594 / 143	0.830 / 208	**1.010 / 237**
		BQSquare	0.080 / 8	0.129 / 18	0.201 / 38	-0.010 / 15	0.337 / 65	0.640 / 125	**0.886 / 141**
		BlowingBubbles	0.164 / 35	0.184 / 58	0.228 / 68	0.386 / 120	0.533 / 170	**0.740** / 226	0.710 / **230**
		BasketballPass	0.259 / 58	0.307 / 75	0.335 / 78	0.628 / 138	0.728 / 155	1.080 / 212	**1.110 / 222**
	E	FourPeople	0.373 / 50	0.388 / 60	0.459 / 70	0.664 / 85	0.734 / 95	0.940 / 117	**1.021 / 136**
		Johnny	0.247 / 10	0.315 / 40	0.378 / 40	0.548 / 55	0.604 / 68	0.810 / 88	**0.976 / 120**
		KristenAndSara	0.409 / 50	0.421 / 60	0.481 / 60	0.655 / 75	0.754 / 85	0.970 / 96	**1.035 / 113**
		Average	0.233 / 45	0.263 / 58	0.300 / 63	0.455 / 88	0.562 / 109	0.830 / 151	**0.925 / 176**

Table 3. Overall comparison of compressed video enhancement for ΔPSNR (dB) and ΔSSIM ($\times 10^{-4}$) over test sequences at QP=37. We experiment with five different video resolutions: A (2,560×1,600), B (1,920×1,080), C (832×480), D (480×240), E (1,280×720).

Figure 5. Visual comparison of video deblurring results on the GOPRO (Nah et al., 2017) dataset. Please zoom in for a better comparison.

(Hyun Kim et al., 2017), Nah *et al.* (Nah et al., 2019b), EDVR (Wang et al., 2019), STFAN (Zhou et al., 2019), TSP (Pan et al., 2020), and UHDVD (Deng et al., 2021). The Tab. 2 shows the quantitative results on the GOPRO dataset (Nah et al., 2017). Our proposed method performs favorably against other methods and has an absolute advantage on PSNR in video deblurring. Specifically, the S2SVR model achieves a performance gain of 0.14dB on the dataset with a lightweight structure. We also report the size of the open-source model in Tab. 2. As the largest model, EDVR's parameter is up to 23M, but its performance is unsatisfactory. Our S2SVR network contains 8.44M parameters. Compared

with the TSP (Pan et al., 2020), our model achieves a higher PSNR performance with only one-half of its size.

Visual Comparison. From the comparison results in Fig. 5, it can be seen that our method can restore the original structure as much as possible from the severely degraded scene. Digital restoration of blurred scenes is difficult. It can be seen that no other method except ours can guarantee the semantics while still retaining the satisfying visual results.

4.4. Compressed Video Enhancement

Quantitative Comparison. We evaluate the performance of compressed video enhancement by ΔPSNR and ΔSSIM, which measure the PSNR and SSIM improvement after the
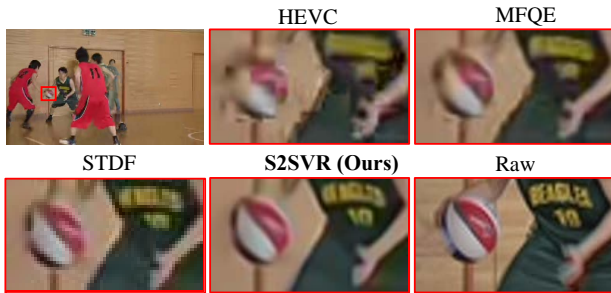
Figure 6. Visual comparison on Video BasketballPass at QP = 37.

(a) Without Motion Compensation
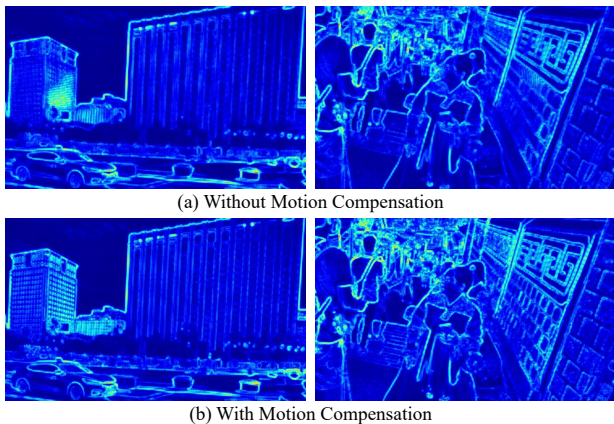

(b) With Motion Compensation

Figure 7. Visualization with and without motion compensation.

enhancement. We compare S2SVR with AR-CNN (Dong et al., 2015), DnCNN (Zhang et al., 2017), DS-CNN (Yang et al., 2018a), MFQE 1.0 (Yang et al., 2018b), MFQE 2.0 (Guan et al., 2019), and STDF-R3L (Deng et al., 2020). As shown in Tab. 3, S2SVR almost outperforms all compared methods in ΔPSNR and ΔSSIM by a large margin. On the $RaceHorses$ dataset with the 832×480 input size, our method can outperform the latest method by at least 0.304dB, which shows the superiority of our method in processing long video sequences with large motion.

Visual Comparison. As shown in Fig. 6, some structural distortions and color deviations make the results of previous methods unconvincing. Our S2SVR network guarantees the basic structural texture and semantic content. Due to space limitations, we put more comparisons in the supplementary.

5. Ablation Study

Unsupervised Optical Flow Estimator. To demonstrate the effectiveness of the unsupervised training scheme, we retrain our optical flow network pwclite in a supervised manner with the optical flow dataset FlyingChairs (Dosovitskiy et al., 2015). We also adopt pre-trained RAFT (Teed & Deng, 2020), the SOTA supervised optical flow network, as our optical flow estimator. As shown in Tab. 4, the pwclite trained with our unsupervised distillation loss can outperform the supervised counterpart by 0.17 dB. It indicates that the flow estimator trained in our unsupervised scheme fits the VR tasks well. Notably, it achieves a better result than the state-of-the-art supervised method RAFT by 0.08dB

Method	RAFT		Pwclite	
	Sup.	Unsup.	Sup.	Unsup.
Params	4.81 M	-	2.24 M	2.24 M
PSNR (dB)	31.88	-	31.79	**31.96**

Table 4. Model analysis with supervised (Sup.) and unsupervised (Unsup.) optical flow estimation model training.

Length	EDVR-M	EDVR	**S2SVR (Ours)**
5	27.78	28.05	28.10 (+ 0.05)
15	28.47	28.80	29.25 (+ 0.45)
30	27.76	28.01	28.53 (+ 0.52)
50	27.52	27.77	28.34 (+ **0.57**)

Table 5. Quantitative comparison on sequences of different length.

with a lower cost. These results show the effectiveness of our unsupervised optical flow method.

Motion Compensation Visualization. We visualize the feature saliency maps with (W/I) and without (W/O) motion compensation in Fig. 7. Obviously, the video frame will lose lots of motion details and texture edges without motion compensation. It is caused by the misalignment among multiple frames, which limits the potential of the seq2seq model in VR. In contrast, with motion compensation, the feature map is much sharper and preserves more movement details, which benefits from our accurate optical flow estimation. Motion compensation narrows the domain difference between NLP and VR, facilitating information aggregation.

Long Sequence Reconstruction. To validate the effectiveness of our S2SVR in capturing long-range temporal dependencies, we separate a video in REDS4 dataset into 4 segments with different lengths, including $5, 15, 30,$ and 50 frames, respectively. And we use S2SVR, EDVR, and EDVR-M to restore these sequences independently. Our method performs the best among the three methods in Tab. 5. And the longer the sequence is, the more superior our S2SVR shows. It suggests that our method has an excellent performance in modeling long-range dependencies.

6. Conclusions

In this paper, we propose an unsupervised flow-aligned seq2seq model for multiple video restoration tasks. Our work aims at solving the challenges of properly modeling the inter-frame relation within the video sequence. The sequence-to-sequence learning is explored for the first time in VR to capture long-term temporal dependencies at a low cost. What's more, we design an unsupervised optical method equipped with a novel distillation loss to improve the performance of the seq2seq model in VR. Extensive experiments show that the proposed method achieves comparable performance in video deblurring, video super-resolution, and compressed video quality enhancement tasks with moderate model size, especially in long sequence VR.

References

Cai, Y., Hu, X., Wang, H., Zhang, Y., Pfister, H., and Wei, D. Learning to generate realistic noisy images via pixel-level noise-aware adversarial training. In *NeurIPS*, 2021.

Cai, Y., Lin, J., Hu, X., Wang, H., Yuan, X., Zhang, Y., Timofte, R., and Van Gool, L. Mask-guided spectral-wise transformer for efficient hyperspectral image reconstruction. In *CVPR*, 2022a.

Cai, Y., Lin, J., Lin, Z., Wang, H., Zhang, Y., Pfister, H., Timofte, R., and Van Gool, L. Mst++: Multi-stage spectral-wise transformer for efficient spectral reconstruction. In *CVPRW*, 2022b.

Cao, J., Li, Y., Zhang, K., and Van Gool, L. Video super-resolution transformer. *arXiv preprint arXiv:2106.06847*, 2021.

Chan, K. C., Wang, X., Yu, K., Dong, C., and Loy, C. C. Basicvsr: The search for essential components in video super-resolution and beyond. In *CVPR*, 2021.

Chen, M. X., Firat, O., Bapna, A., Johnson, M., Macherey, W., Foster, G., Jones, L., Parmar, N., Schuster, M., Chen, Z., et al. The best of both worlds: Combining recent advances in neural machine translation. In *ACL*, 2018.

Chopra, S., Auli, M., and Rush, A. M. Abstractive sentence summarization with attentive recurrent neural networks. In *NAACL-HLT*, 2016.

Dai, Q., Yoo, S., Kappeler, A., and Katsaggelos, A. K. Dictionary-based multiple frame video super-resolution. In *ICIP*, 2015.

Deng, J., Wang, L., Pu, S., and Zhuo, C. Spatio-temporal deformable convolution for compressed video quality enhancement. In *AAAI*, 2020.

Deng, S., Ren, W., Yan, Y., Wang, T., Song, F., and Cao, X. Multi-scale separable network for ultra-high-definition video deblurring. In *CVPR*, 2021.

Dong, C., Deng, Y., Loy, C. C., and Tang, X. Compression artifacts reduction by a deep convolutional network. In *ICCV*, 2015.

Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., and Brox, T. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015.

Godard, C., Mac Aodha, O., and Brostow, G. J. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

Guan, Z., Xing, Q., Xu, M., Yang, R., Liu, T., and Wang, Z. Mfqe 2.0: A new approach for multi-frame quality enhancement on compressed video. *TPAMI*, 2019.

Haris, M., Shakhnarovich, G., and Ukita, N. Recurrent back-projection network for video super-resolution. In *CVPR*, 2019.

He, D., Zheng, Y., Sun, B., Wang, Y., and Qin, H. Checkerboard context model for efficient learned image compression. In *CVPR*, 2021.

Hyun Kim, T., Mu Lee, K., Scholkopf, B., and Hirsch, M. Online video deblurring via dynamic temporal blending network. In *ICCV*, 2017.

Isobe, T., Jia, X., Gu, S., Li, S., Wang, S., and Tian, Q. Video super-resolution with recurrent structure-detail network. In *ECCV*, 2020.

Jo, Y., Oh, S. W., Kang, J., and Kim, S. J. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *CVPR*, 2018.

Jozefowicz, R., Zaremba, W., and Sutskever, I. An empirical exploration of recurrent network architectures. In *ICML*, 2015.

Kuznetsov, V. and Mariet, Z. Foundations of sequence-to-sequence modeling for time series. In *AISTATS*, 2019.

Li, W., Tao, X., Guo, T., Qi, L., Lu, J., and Jia, J. Mucan: Multi-correspondence aggregation network for video super-resolution. In *ECCV*, 2020.

Liang, J., Cao, J., Fan, Y., Zhang, K., Ranjan, R., Li, Y., Timofte, R., and Van Gool, L. Vrt: A video restoration transformer. *arXiv preprint arXiv:2201.12288*, 2022.

Liao, R., Tao, X., Li, R., Ma, Z., and Jia, J. Video super-resolution via deep draft-ensemble learning. In *ICCV*, 2015.

Lin, J., Cai, Y., Hu, X., Wang, H., Yan, Y., Zou, X., Ding, H., Zhang, Y., Timofte, R., and Van Gool, L. Flow-guided sparse transformer for video deblurring. *arXiv preprint arXiv:2201.01893*, 2022a.

Lin, J., Cai, Y., Hu, X., Wang, H., Yuan, X., Zhang, Y., Timofte, R., and Van Gool, L. Coarse-to-fine sparse transformer for hyperspectral image reconstruction. *arXiv preprint arXiv:2203.04845*, 2022b.

Liu, L., Zhang, J., He, R., Liu, Y., Wang, Y., Tai, Y., Luo, D., Wang, C., Li, J., and Huang, F. Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation. In *CVPR*, 2020.

Mémin, E. and Pérez, P. Dense estimation and object-based segmentation of the optical flow with robust techniques. *TIP*, 1998.

Nah, S., Hyun Kim, T., and Mu Lee, K. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, 2017.

Nah, S., Baik, S., Hong, S., Moon, G., Son, S., Timofte, R., and Mu Lee, K. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *CVPRW*, 2019a.

Nah, S., Son, S., and Lee, K. M. Recurrent neural networks with intra-frame iterations for video deblurring. In *CVPR*, 2019b.

Ohm, J.-R., Sullivan, G. J., Schwarz, H., Tan, T. K., and Wiegand, T. Comparison of the coding efficiency of video coding standards—including high efficiency video coding (hevc). *TCSVT*, 2012.

Ott, M., Edunov, S., Grangier, D., and Auli, M. Scaling neural machine translation. *arXiv preprint arXiv:1806.00187*, 2018.

Pan, J., Bai, H., and Tang, J. Cascaded deep video deblurring using temporal sharpness prior. In *CVPR*, 2020.

Sajjadi, M. S., Vemulapalli, R., and Brown, M. Frame-recurrent video super-resolution. In *CVPR*, 2018.

Shahar, O., Faktor, A., and Irani, M. *Space-time super-resolution from a single video*. IEEE, 2011.

Shi, T., Keneshloo, Y., Ramakrishnan, N., and Reddy, C. K. Neural abstractive text summarization with sequence-to-sequence models. *ACM Transactions on Data Science*, 2021.

Shi, X., Gao, Z., Lausen, L., Wang, H., Yeung, D.-Y., Wong, W.-k., and Woo, W.-c. Deep learning for precipitation nowcasting: A benchmark and a new model. In *NeurIPS*, 2017.

Su, S., Delbracio, M., Wang, J., Sapiro, G., Heidrich, W., and Wang, O. Deep video deblurring for hand-held cameras. In *CVPR*, 2017.

Sun, D., Yang, X., Liu, M.-Y., and Kautz, J. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*, 2014.

Takeda, H., Milanfar, P., Protter, M., and Elad, M. Super-resolution without explicit subpixel motion estimation. *TIP*, 2009.

Tao, X., Gao, H., Shen, X., Wang, J., and Jia, J. Scale-recurrent network for deep image deblurring. In *CVPR*, 2018.

Teed, Z. and Deng, J. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020.

Tian, Y., Zhang, Y., Fu, Y., and Xu, C. Tdan: Temporally-deformable alignment network for video super-resolution. In *CVPR*, 2020.

Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T., and Saenko, K. Sequence to sequence-video to text. In *ICCV*, 2015.

Wang, X., Chan, K. C., Yu, K., Dong, C., and Change Loy, C. Edvr: Video restoration with enhanced deformable convolutional networks. In *CVPRW*, 2019.

Wang, Y., Yang, Y., Yang, Z., Zhao, L., Wang, P., and Xu, W. Occlusion aware unsupervised learning of optical flow. In *CVPR*, 2018a.

Wang, Y., Yang, Y., Yang, Z., Zhao, L., Wang, P., and Xu, W. Occlusion aware unsupervised learning of optical flow. In *CVPR*, 2018b.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004.

Wedel, A., Cremers, D., Pock, T., and Bischof, H. Structure- and motion-adaptive regularization for high accuracy optic flow. In *ICCV*, 2009.

Xiang, X., Wei, H., and Pan, J. Deep video deblurring using sharpness features from exemplars. *TIP*, 2020.

Xue, T., Chen, B., Wu, J., Wei, D., and Freeman, W. T. Video enhancement with task-oriented flow. *IJCV*, 2019.

Yang, R., Xu, M., Liu, T., Wang, Z., and Guan, Z. Enhancing quality for hevc compressed videos. *TCSVT*, 2018a.

Yang, R., Xu, M., Wang, Z., and Li, T. Multi-frame quality enhancement for compressed video. In *CVPR*, 2018b.

Yang, R., Sun, X., Xu, M., and Zeng, W. Quality-gated convolutional lstm for enhancing compressed video. In *ICME*, 2019.

Yi, P., Wang, Z., Jiang, K., Jiang, J., and Ma, J. Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations. In *ICCV*, 2019.

Yu, J. J., Harley, A. W., and Derpanis, K. G. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *ECCV*, 2016.

Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *TIP*, 2017.

Zheng, H., Li, X., Liu, F., Jiang, L., Zhang, Q., Li, F., Dang, Q., and He, D. Adaptive spatial-temporal fusion of multi-objective networks for compressed video perceptual enhancement. In *CVPR*, 2021.

Zhong, Z., Gao, Y., Zheng, Y., and Zheng, B. Efficient spatio-temporal recurrent neural network for video deblurring. In *ECCV*, 2020.

Zhou, S., Zhang, J., Pan, J., Xie, H., Zuo, W., and Ren, J. Spatio-temporal filter adaptive network for video deblurring. In *ICCV*, 2019.