Gated Spatio-Temporal Attention-Guided Video Deblurring

Maitreya Suin A. N. Rajagopalan
Indian Institute of Technology Madras, India
maitreyasuin21@gmail.com, raju@ee.iitm.ac.in

Abstract

Video deblurring remains a challenging task due to the complexity of spatially and temporally varying blur. Most of the existing works depend on implicit or explicit alignment for temporal information fusion, which either increases the computational cost or results in suboptimal performance due to misalignment. In this work, we investigate two key factors responsible for deblurring quality: how to fuse spatio-temporal information and from where to collect it. We propose a factorized gated spatio-temporal attention module to perform non-local operations across space and time to fully utilize the available information without depending on alignment. First, we perform spatial aggregation followed by a temporal aggregation step. Next, we adaptively distribute the global spatio-temporal information to each pixel. It shows superior performance compared to existing non-local fusion techniques while being considerably more efficient. To complement the attention module, we propose a reinforcement learning-based framework for selecting keyframes from the neighborhood with the most complementary and useful information. Moreover, our adaptive approach can increase or decrease the frame usage at inference time, depending on the user's need. Extensive experiments on multiple datasets demonstrate the superiority of our method.

1. Introduction

Video deblurring, as a primary problem in the vision and graphics communities, strives to predict latent frames from a blurred sequence. The camera shake and high-speed motion in dynamic scenes often generate unwanted blur and produce blurry videos. Such videos not only deteriorate the visual quality but also hinder some high-level vision tasks such as tracking [13, 18], video stabilization [17], etc. As more videos are taken using hand-held and onboard video capturing devices, this problem has received great attention in the last decade. The blur in videos is usually a consequence of several interwoven factors like camera shake, object motion, depth variations, etc.

Unlike single-image deblurring, video deblurring methods can utilize additional information that exists across neighboring frames. Early methods relied on motion compensation of the input frames, either explicitly [24, 16, 12] or implicitly [37, 36], to aggregate information at a particular location from adjacent frames. [24, 12, 3] first compute optical flow between a reference frame and neighboring frames and then use the aligned observations to deblur the reference frame. [32] utilizes deformable convolution to align feature maps using learnable offsets. Although these alignment methods are intended for increasing the temporal coherence, they have several disadvantages: 1) they introduce extra parameters, calculations and training difficulty, and 2) incorrect alignment may lead to undesired artifacts. Implicit handling of motion using recurrent networks or 3D convolution has its own drawbacks. 3D convolution [35] is computationally heavy and introduces a large number of parameters. For recurrent architectures, the assumption that all previous frames will be automatically aligned and fused in the hidden state remains a problem for frames with large displacement. It is not very easy to extract only the relevant information from a single combined state. Also, due to recurrent connections it is not feasible to process multiple frames in parallel.

In this work, we address two critical aspects of video deblurring: how to gather spatio-temporal information effectively and from where to gather this information. The key intuition is that a blurred region in the current frame would probably have complementary information in a distant frame. Finding the spatio-temporal relation is critical while fusing information as not all parts of the neighboring frames are equally informative for restoring the current frame due to varying factors such as occlusions, motion, etc. Fusion of incorrect information adversely affects reconstruction performance. We explore the need for non-local operations for spatio-temporal fusion. A non-local self-attention module aims at computing the correlations between all possible pixels within and across frames, which directly resonates with the current goal of spatio-temporal fusion. By nature, such a block does not require any alignment steps. [30] introduced self-attention based transformer

network for natural language processing, [33] showed a similar non-local approach for classification and recognition. However, extending such approaches to generation tasks is non-trivial. Despite its exceptional non-local processing capabilities, even simpler spatial self-attention can be hard to implement due to its large memory requirement for the image domain. For spatio-temporal operation in videos, it will become significantly more expensive. [33] downsamples the input by a large scale factor, even to 7×7 feature maps. But, for generation tasks, where every pixel accuracy matters, such downsampling will be harmful. For general video deblurring, the input size can be arbitrary and quite large.

In this paper, we present a factorized spatio-temporal self-attention mechanism that contains the essential properties of non-local processing in spatio-temporal domain while being much more efficient. We formulate the entire non-local operation as the composition of three lightweight operators: spatial aggregation, temporal aggregation, and pixelwise adaptive distribution. It requires significantly less memory (almost 90% for $5 \times 128 \times 128$ patch) compared to existing non-local blocks for the same spatio-temporal size while providing superior performance. Further, we incorporate feature-gating to put more focus on aggregating sharp features from the neighborhood. We discuss the details in Section 3 and advantages over existing video deblurring methods in Section 2.

Next, to complement the spatio-temporal attention module, we delve deeper into a rather unexplored area of video restoration - finding the temporal locations to fuse information from. Earlier works use the immediate neighborhood of the current frame (for example, ± 2 frames) or sequential frame processing for recurrent methods. These approaches assume that the immediate vicinity will contain all the required information for restoring the current one. Applying a fixed neighborhood to all frames is a sub-optimal design choice. Each frame to be deblurred has a distinct appearance, and different parts of the frame can have complementary information beyond the typical fixed temporal window. As shown in the example in Fig. 1, the $(t+7)^{th}$ frame is the most useful one for deblurring the text whereas $(t+6)^{th}$ frame contains sharper features of the person. Therefore, we could focus on these frames while skipping the unnecessary ones. Albeit being intuitive, the simple solution of extending the neighborhood size in existing works will hardly solve the problem. On the one hand, it will significantly increase the amount of computations, and on the other, distant frames will create more issues due to misalignment and fusion of wrong information. Also, depending on the severity of the blur in the current frame, we would ideally want to look into a varying number of frames for efficient utilization of the available computing resources.

In contrast to the commonly used one-size-fits-all scheme, we would like to make these decisions individually per input frame. Based on this intuition, we present a new perspective for video deblurring by deciding on-the-fly which frames from the neighborhood to use on a per-frame basis. Empirically, we set a maximum temporal window, beyond which we observe that the scene content changes significantly to be of any use. We train a lightweight reinforcement learning agent (referred to as the frame selection network) to pick a certain number of keyframes within this large window. We design a novel reward function that allows the agent to look for more frames for severely blurred frames and skip unnecessary processing for easier ones. Further, to adapt to different applications, we design the frame selection network to take input from the user at inference time and increase or decrease the number of neighboring frames usage adaptively while providing the best possible restoration performance.

Figure 1. Varying amount of blur across frames.

To summarize, our contributions are

- We introduce a factorized spatio-temporal attention as an effective non-local information fusion tool for video deblurring task.
- To the best of our knowledge, our work is the first to present an approach for finding the key-frames with the most relevant information for video deblurring. It significantly boosts the restoration performance when coupled with the proposed attention module.
- Extensive experiments and analysis are presented on several video deblurring benchmarks to show state-of-the-art accuracy and interpretability achieved by our architecture.

1.1. Language

2. Related Works

Early video or multiframe deblurring methods [4, 17] usually assume that there exist sharp contents and interpolate them to help the restoration of latent frames. The main success of these methods is due to the use of sharp contents from adjacent frames. Recently, several end-to-end CNN methods [24, 11, 26, 25] have been proposed for image or video deblurring. [35] employ 3D convolutions to help latent frame restoration. [10] treat optical flow as a line-shaped approximation of blur kernels, which optimize optical flow and blur kernels iteratively. [11] develop a

spatial-temporal recurrent network with a dynamic temporal blending layer, where they concatenated feature of the current frame and the previous frame and pass through a recurrent network. [37] fed the previous deblurred frame along with the current blurry frame through their network in a progressive manner and modeled frame alignment and non-uniform blur removal as element-wise filter adaptive convolution processes. [32] develop pyramid, cascading, and deformable convolution to achieve better alignment performance. They have used a simpler temporal and spatial attention strategies. First, they align the neighboring frames, and then at each pixel location, they aggregate the information using convolution. For spatial attention, they have used simple mask multiplication. In comparison, we resort to more effective non-local processing [1, 33] using the proposed gated spatio-temporal attention module where each pixel in the current frame can gather complementary information from all other pixels in all the frames. [21] proposed a cascaded deblurring approach while utilizing temporal sharpness prior. Most of these works depend on either explicit or implicit alignment process. None of the previous deep learning-based works explored the usefulness of finding the most complementary frames from the neighborhood to the best of our knowledge. Our work focuses on these two key factors: a) Which frames are most helpful b) How to fuse relevant information from those frames effectively.

3. Method

An overview of our network is shown in Fig. 2. We use an encoder-decoder architecture comprising of densely connected modules as the backbone of our restoration network. At the encoder, we use spatio-temporal self-attention blocks to fuse features of the current frame and the selected neighboring frames. The neighboring frames are stacked along the batch dimension and passed through initial levels of the encoder for feature extraction. Similarly, in decoder, we use skip connection to fuse the neighboring frames features with the decoder features of the current frame. The attention module (Fig. 3) is used inside a residual block, which consists of convolutional layers, Batch-Norm, and ReLU layers. We use a stack of few lightweight convolutional layers, a fully connected layer followed by sigmoid as the frame-selection network (FSN). For deblurring a particular frame, we concatenate all the neighboring frames within a temporal window and pass through the lightweight FSN. The FSN decides which frames contain the most useful information. Only those frames, along with the reference frame, are then fed to the restoration network. Note that the FSN is very lightweight in nature. It introduces only 0.2 M extra parameters while significantly improving the performance. More analysis is given in the experimental section, and the layerwise description of the architecture is provided in the supplementary document. Next, we describe each building block of our proposed video deblurring network and explain how they come together to solve the task at hand.

3.1. Gated Spatio-Temporal Attention:

Non-local means [1] is a classical filtering algorithm that allows distant pixels to contribute to the filtered response at a location based on patch appearance similarity. This non-local filtering idea was later developed into block-matching algorithm, which was used with neural networks for image denoising [14]. A similar technique was shown to be successful in the natural language processing domain [30]. The main building block of [30] is a self-attention module that computes the response at a position in a sequence (e.g., a sentence) by attending to all positions and taking their weighted average in an embedding space. [33] proposed a generic non-local operation in deep neural networks to calculate the relation between all possible positions. Given an input feature map of size $T \times H \times W$ (omitting the channel dimension for brevity), the goal of non-local block is to compute the relation $THW \times THW$. But, the tensor of size $THW \times THW$ is huge for videos and to reduce the computational overhead, [33] typically used $T = 4$, $H = W = 7$. Further, [33] resort to sub-sampling trick or reduced number of channels for some cases.

For a restoration task like video deblurring, large down-sampling will deteriorate pixel-level accuracy. Some works like [23] use non-local operations in small blocks inside an image, which hinders its expressibility. Instead, we propose a factorized spatio-temporal self-attention module, which has two significant advantages: 1) it can gather global information for each pixel without requiring any explicit down-sampling, 2) owing to its design, this module does not need any alignment operation to be performed. Our design is intuitively motivated by examining the flow of information in [9, 2, 15] etc, which deploy a squeeze-based aggregation operation in their approach. We gather global information from spatial and temporal domain by performing squeezing operation, and then adaptively distribute it to each pixel of the current frame. We construct three lightweight operations, including spatial squeezing, temporal squeezing, and pixelwise adaptive distribution.

Spatial Aggregation: For simplicity, we assume batch and channel dimensions to be 1 in the following sections, but it can have any standard values. Given an input tensor $x \in \mathbb{R}^{T \times 1 \times HW}$, we calculate a set of spatial attention maps as

$$A_s = \text{softmax}_{HW}(f_s(x)) \tag{1}$$

where $A_s \in \mathbb{R}^{T \times M \times HW}$, f_s is convolutional operation, softmax_{HW} is softmax along HW and M is the number of attention maps per frame. Note that we transfer each frame (T) to batch dimension; so the attention map calculations are performed for each frame separately. Thus, the attention maps corresponding to each frame are automatically

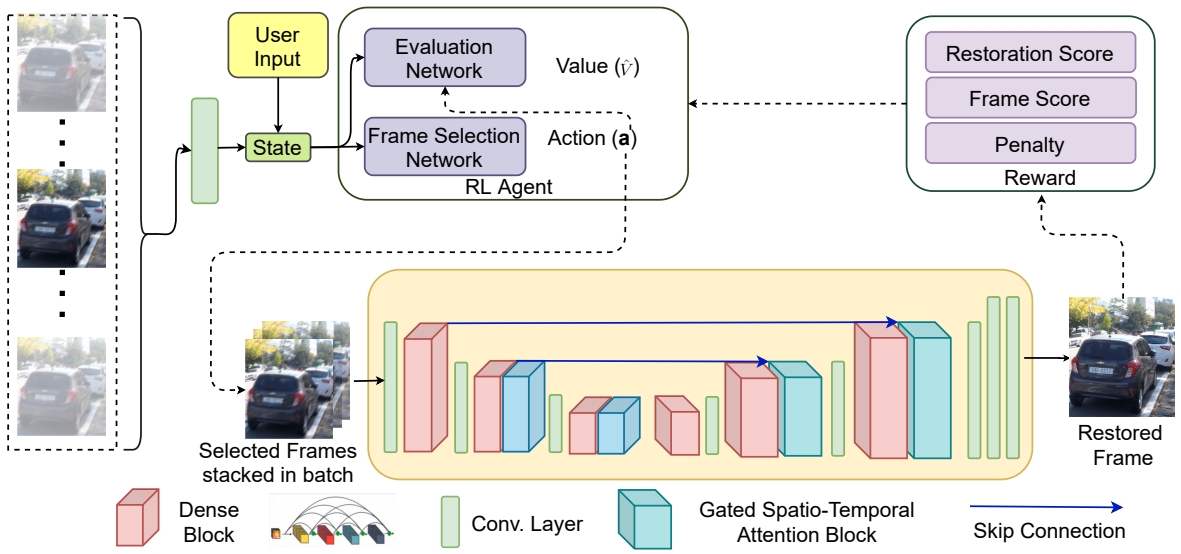

Figure 2. An overview of our method.

aligned with their input frames. Intuitively, it generates M number of spatial attention maps for each frame. Next, we elementwise multiply each frame with each of these M attention maps. Let, $A_s^m \in \mathbb{R}^{T\times HW}$ denote the m^{th} attention map. Non-local spatial feature is aggregated for each of the frames using the m^{th} attention map as

$$G_s^m = \mathrm{S}_{HW}(A_s^m \odot x) \tag{2}$$

where $G_s^m \in \mathbb{R}^T$, $m \in 1, ..., M$ and S_{HW} represents the squeeze operation [9] along HW.

Temporal Aggregation: Now, we calculate a set of temporal attention maps as

$$A_t = \mathrm{softmax}_T(f_t(x')) \tag{3}$$

where $A_t \in \mathbb{R}^{N\times T}$, f_t is 1D convolutional operator, $x' \in \mathbb{R}^{1\times T}$ is the spatially pooled version of the input feature x, $\mathrm{softmax}_T$ is softmax operation along T. Given the set of temporal attention maps A_t and the spatially aggregated features G_s, we apply the N temporal attention maps on each of the $G_s^m : m \in 1, ..., M$ and aggregate temporal information as

$$G_{st} = G_s A_t^{Tr} \tag{4}$$

where $G_{st} \in \mathbb{R}^{MN}$, $G_s \in \mathbb{R}^{M\times T}$, Tr represents transpose operation. Intuitively, each of these MN elements contains global spatio-temporal information, which has been aggregated using the factorized M spatial attention maps and N temporal attention maps resulting in a total of MN possible combinations.

Pixelwise Adaptive Distribution: After aggregating global information, we adaptively distribute it to each pixel. We generate a pixelwise attention map A_p as

$$A_p = \mathrm{softmax}_1(f_p(x^R)) \tag{5}$$

where $A_p \in \mathbb{R}^{MN\times HW}$, f_p is 2D convolutional operation, $x^R \in \mathbb{R}^{1\times HW}$ is the feature map of the current frame. Each pixel will adaptively selectly a particular combination of total MN spatio-temporal attention map using A_p. A similar intuition can be found in [15], which finds a compact basis set for iterative expectation-maximization approach. Now, we distribute the global information to each pixel as

$$y^R = G_{st} A_p \tag{6}$$

where $y^R \in \mathbb{R}^{1\times HW}$ is the output feature map corresponding to the current frame. For C input channels, y^R will expand to $(C \times HW)$ tensor. We eliminated the need for a huge $THW \times THW$ matrix through this factorized processing, and all these tensor operations can be efficiently implemented in modern deep learning libraries, making it both fast and memory efficient.

Gating Operation: Owing to the nature of non-local operations, the attention module will allow each pixel to look for similar information present across the spatio-temporal neighborhood. To encourage the gathering of sharper information for aiding the restoration of blurry reference frame, we use a blur-mask to explicitly give more weightage to similar and sharp features while performing the squeezing operation. A similar observation was mentioned recently in [21], where the pixel intensities were compared among neighboring frames to decide if the pixel is sharp. But, this approach will fail if the nature of blur is similar in consecutive frames. Also, [21] resorts to simple concatenation of the mask before feeding to convolutional layers. Instead, we use a single convolutional layer followed by a sigmoid to predict the blur mask Q as

$$Q = \mathrm{sigmoid} f_{conv}(x) \tag{7}$$

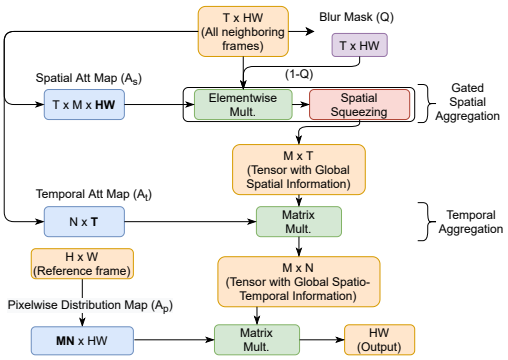

Figure 3. Gated Spatio-Temporal Attention. Bold denotes output after softmax.

Value 1 represents a blurry pixel and 0 otherwise. Next, we explicitly use this as a gating function to give more weightage to sharper features while performing spatial aggregation. Eq. 2 can be modified as

$$G_s^m = \mathrm{S}_{HW}(A_s^m \odot x \odot (1 - Q)) \tag{8}$$

We force Q to match the ground-truth mask. Akin to [22, 31], we took the difference between the blurry and sharp frames and then applied thresholding to get the ground-truth blur mask.

3.2. Frame-Selection Strategy:

We formulate key-frames selection as a decision making problem which naturally fits into a reinforcement learning framework. We design two networks: a frame-selection network (FSN) and an evaluation network (EN). FSN can be viewed as a reinforcement learning agent that takes the current frame, and its neighboring frames inside T_{max} (the maximum temporal window) as input and generates a set of actions to decide which frames to watch. The agent's goal is to derive an effective frame selection strategy that achieves maximum restoration accuracy while using as few frames as possible. EN is used for training the FSN and not used at inference time. The final restoration performance for the current frame, along with the evaluation network's output, steers the frame selection network to pick only the frames which contain the most complementary information.

We model this problem as a multi-agent reinforcement learning problem. The number of agents is T_{max}, where each agent takes a decision for the corresponding frame. Recently, [7] used a multi-agent framework, where they have used a separate pixelwise reward for each agent. For our scenario, we will have a single external reward, which is the final restoration accuracy for the current frame. [5, 27] also adopted final accuracy as their reward. Although this global reward can be used to train our multi-agent framework, it will be difficult to converge to the optimal policy. Instead, following [6], we exploit EN to give different rewards to different agents depending on how useful their action was. EN is trained to predict the expected reward from the current frame selection made by FSN. The whole process is described in detail next.

State Variable: The state variable s acts as the input to FSN. It has two parts. First, we generate x_{cat}, which represents concatenation of all the frames in the temporal window T_{max} and pass it through a few convolutional layers. Then, we apply pooling along the spatial and channel dimensions to get a T dimensional vector p_{cat}. We have another user-defined variable u, which is encoded in the state to let the agent know the maximum number of neighboring frames it is expected to pick. The final state variable is $s = [p_{cat}; u]$.

Action Space: The frame-Selection Network accepts the state variable (s) and maps it into a policy that provides the probability of different actions to be exercised for all the agents. For our case, the action space is binary with two actions: 1: Pick. 0: Skip. A few possible alternatives can be to use a single agent that outputs probabilities for all possible $2^{T_{max}}$ set of actions or to divide this problem into T_{max} independent sub-problems and train T_{max} separate networks. Both of these options are computationally sub-optimal. To handle this, we employ a shared network among all the T_{max} agents, and we can parallelize the computation on a GPU, which makes it efficient.

Frame Selection Network (FSN): The FSN is made of a fully connected layer which is parameterized by θ_{fs}, followed by a sigmoid. It produces the policy $\pi_a(\cdot|s; \theta_{fs})$, which gives the probability distribution of the actions to be taken i.e. which frames to pick. The actions are sampled from π_a during training. During the inference, we use maximum aposteriori estimation to choose the actions $\mathbf{a} = \arg\max_{a \in 0,1}(\pi_a(\cdot|s; \theta_{fs})) \in \mathbb{R}^{T_{max}}$. Note that, for some situation if T_{max} is variable, the fully connected layer can be replaced with a 1D convolutional layer to handle inputs of arbitrary length.

Reward Function: The reward function reflects how good are the actions taken by FSN in picking neighboring frames. We introduce a reward function that not only helps increase the restoration quality but encourages skipping redundant frames. Our reward function can be described as

$$r = \alpha_1 \cdot \frac{RS}{P} + \alpha_2 \cdot FS \tag{9}$$

where RS, FS, P are restoration score, frame score, and penalty, respectively. The first term is the ratio of improvement in PSNR value (RS) to the number of frames fused (P). It encourages the agent to skip frames if it does not improve the restoration. FS is the regularizing term that controls the maximum number of frames that can be watched. FS is 1 if the number of picked frames is less than the maximum (u) and 0 otherwise. This term forces the agent to be aware of

the information u present in the state variable and adjust its behavior accordingly.

Evaluation Network (EN): We use the EN to assess the actions taken by FSN. EN has a fully connected layer f_g, which is parameterized by θ_g. It takes the state variable s and the set of actions $\mathbf{a}$ as input and produces an output $f_g([\mathbf{s}, \mathbf{a}]; \theta_g) = \hat{V}$ which is known as the value function [28]. The job of EN is to approximate the value function, i.e., expected reward from the current state ($\mathbb{E}_{\mathbf{s},\mathbf{a}} r$). The actual reward, acquired from empirical rollouts, is then compared to the value predicted by EN and used to update frame-selection network parameters in the direction of performance improvement.

Reinforcement learning Loss: Reinforcement loss also has two parts. The evaluation network is trained with the following regression loss

$$\mathcal{L}_G(\theta_g) = \frac{1}{2}||\hat{V} - r||_2 \tag{10}$$

Now, we formulate a separate reward for each agent. One naive solution is to run the restoration network repeatedly after changing a particular action. For example, lets say the i^{th} agent has produced output 1 (pick the i^{th} frame). We change this action to 0 (remove the frame), perform deblurring, and calculate the new reward. Now, the difference between the new reward and the old reward can be treated as how important the i^{th} frame was (consequently, how correct the original decision was). But doing this for all the agents for each frame is practically infeasible due to an excessive amount of training computation. Instead, we exploit the EN to generate separate rewards, learned directly from the agents' experiences instead of relying on extra simulations.

The intuition is that EN can be used to reason about the effect of changing an action. EN is trained to predict the expected reward given the state and the actions taken, so we can change a particular action a_i and feed it to EN to get the expected reward without actually performing the operation. Then, we compute an advantage function that compares the value function for a particular action a_i where $i \in 1, 2, ..., T_{max}$, to a counterfactual baseline that marginalises out a_i while keeping the other agents' actions fixed

$$W^i(\mathbf{s}, \mathbf{a})) = V(\mathbf{s}, \mathbf{a}) - \sum_{a'_i} (\pi(a_i|s)\hat{V}(\mathbf{s}, (\mathbf{a}^{-i}; a'_i)) \tag{11}$$

The policy gradient for the FSN can be expressed as

$$\nabla = \mathbb{E}_\pi \left[\sum_i \nabla_\Theta \log \pi(a_i|s) W^i(\mathbf{s}, \mathbf{a})) \right] \tag{12}$$

For a detailed analysis of the policy gradient technique we encourage the reader to refer to [28].

Training Strategies: Generally, for training an RL agent, we need a stable environment that can provide reasonable rewards. Thus, we first pre-train our restoration network without the RL agent with a fixed neighborhood of ± 2. Then, we introduce the RL agent and train it to pick a maximum of 5 frames from the large neighborhood ($T_{max} = \pm 12$) adaptively. We compare this model with earlier works for a fair comparison. Further, we tune the parameter u in the state variable, which allows our network to take input from the user at inference time and dynamically change the number of frame usage for a video. More analysis is provided in Sec. 5.

4. Experimental Results

Implementation Details: We compare our model with existing works on DVD [24] and GOPRO dataset [19] under the standard training and testing settings of previous state-of-the-art methods [21, 20]. The size of training patch is 256×256 with minibatch size of 8. We use the ADAM optimizer learning rate of $1e^{-4}$, which decreases to half after every 200 epochs. We implement our algorithm based on the PyTorch on an Ubuntu 16 system, Intel Xeon E5 CPU, and an NVIDIA Titan Xp GPU.

Quantitative Comparisons: To evaluate the performance of the proposed algorithm, we compare it against the following state-of-the-art algorithms: Tao et al. [29], Su et al. [24], Wieschollek et al. [34], Kim et al. [11], Nah et al. [20], EDVR [32], STFAN [37] and TSP [21]. Tables 1 and 2 show the quantitative results, where the proposed algorithm performs favorably against the state-of-the-art methods in terms of PSNR and SSIM.

Quantitative Comparisons: Figs. 4 and 5 show some deblurred results from the testset of [24] and [19], respectively. We observe that the results of prior works suffer from incomplete deblurring or artifacts. In contrast, our network is able to restore scene details more faithfully, which are noticeable in the regions containing text, edges, etc.

On both the datasets, the proposed method achieves consistently better PSNR, SSIM, and visual results.

Real Video: We further evaluate our algorithm on the real video deblurring dataset by [4]. As shown in Fig. 6, our algorithm generates much clearer frames with better-detailed structures. For example, the text 'friendship' on the two books are much clearer.

5. Network Analysis

We perform the following experiments, as reported in Table 3 on GOPRO dataset. Net1: Backbone encoder-decoder architecture. Net2: Net1 + proposed spatio-temporal attention without gating. Net3: Net1 + proposed gated spatio-temporal attention. Net4: Net3 + proposed frame selection network (FSN). Note that, for a fair com-

Table 1. Quantitative evaluations on the DVD dataset [24] in terms of PSNR and SSIM. $*$ denotes the reported results from [21].

Methods	Kim and Lee [10]	Gong et al. [8]	Tao et al. [29]	Su et al. [24]	Kim et al. [11]	EDVR [32]*	STFAN [37]	TSP [21]*	Ours
PSNRs	26.94	28.27	29.98	30.01	29.95	28.51	31.15	32.13	**32.53**
SSIMs	0.8158	0.8463	0.8842	0.8877	0.8692	0.8637	0.9049	0.9268	**0.9468**

Table 2. Quantitative evaluations on the GOPRO dataset [19] in terms of PSNR and SSIM. $*$ denotes the reported results from [20, 21].

Methods	Tao et al. [29]	Su et al. [24]	Wieschollek et al. [34]*	Kim et al. [11]*	Nah et al. [20]*	EDVR [32]*	STFAN [37]*	TSP [21]	Ours
PSNRs	30.29	27.31	25.19	26.82	29.97	26.83	28.59	31.67	**32.10**
SSIMs	0.9014	0.8255	0.7794	0.8245	0.8947	0.8426	0.8608	0.9279	**0.96**

(a) Blurred frame (b) GT Frame (c) Tao et al. (d) Su et al. (e) EDVR (f) STFAN (g) TSP (h) Ours

Figure 4. Deblurred results on DVD dataset [24].

(a) Blurred frame (b) GT Frame (c) Tao et al. (d) Su et al. (e) EDVR (f) STFAN (g) TSP (h) Ours

Figure 5. Deblurred results on GOPRO dataset [19].

(a) Blurred frame (b) Kim and Lee (c) STFAN (d) EDVR (e) TSP (f) Ours

Figure 6. Deblurred results on a real video from [4].

Table 3. Network Analysis on GOPRO dataset. STA, GSTA, ± 2, FSN, NLN_A represents spatio-temporal attention, gated spatio-temporal attention, fixed neighborhood of ± 2, using RL based FSN to pick frames, non-local attention of [33], respectively.

	STA	GSTA	± 2	FSN	NLN_A	PSNR
Net1			✓			30.70
Net2	✓		✓			31.40
Net3		✓	✓			31.61
Net4		✓		✓		32.10
Net5				✓	✓	31.17

parison with our final model Net4, we have added the same number of parameters in the baseline to compensate for the few layers of FSN. Further, to verify the effectiveness of the proposed attention module for video deblurring, we replace our attention block with the one used in NLN [33] in Net5. Net2 and Net3 achieve significant improvement over Net1 even when fixed neighboring frames are fused (± 2). It shows the efficacy of the proposed non-local spatio-temporal attention module for video deblurring. We visual-

Table 4. Variation of #Frames at Inference. Rand() denotes picking random frames, FSN(#) denotes picking frames using FSN where it is expected to pick a maximum of '#' frames.

	± 2	Rand(5)	FSN(3)	FSN(5)	FSN(7)
PSNR	31.61	31.65	31.87	32.10	32.19

ize the pairwise self-attention map for Net3 in Fig. 7 for an intuitive understanding of the attention module. As we can see, for different blurry regions, the network is able to look for similar but sharper information in selected neighboring frames, which justifies the effectiveness of the non-local attention module coupled with the frame-selection network for the video deblurring task. For instance, for the image in the first row, we can observe the blurred region with the cyclist is gathering sharper information from $(t-7)^{th}$ frame. On the other hand, the blurry tree region is gathering sharper information from all possible spatial locations in $(t+8)^{th}$ frame. Note that the proposed attention module is able to successfully search for relevant information with-

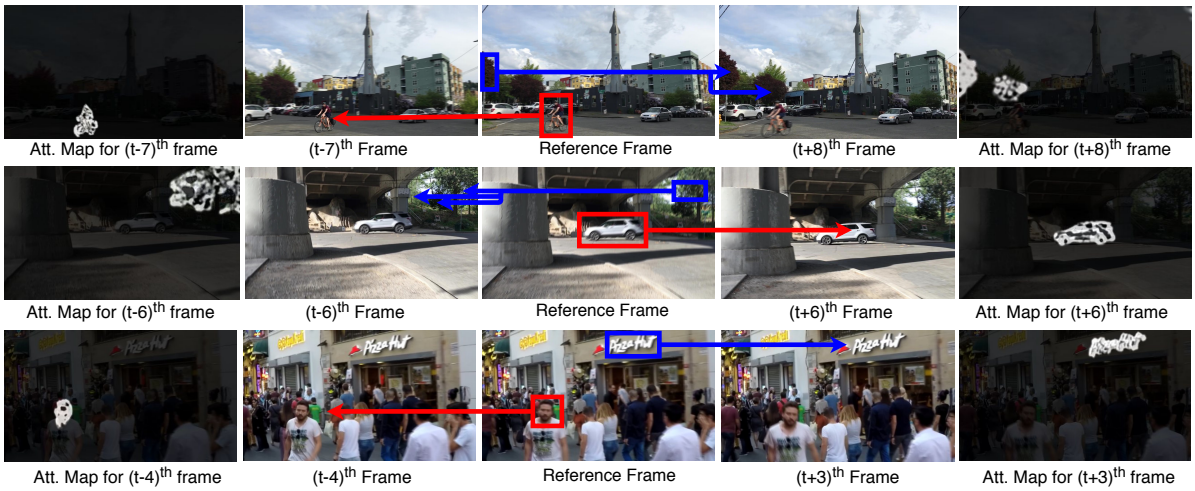

Figure 7. Visualization of selected neighboring frames and corresponding attention map.

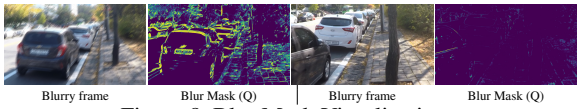

Figure 8. Blur Mask Visualization.

Table 5. Memory Requirement (in GB) of a single attention modules for batch size 1 and varying patch size. *OOM* denotes out-of-memory on a single TitanX GPU.

Patch	16	32	64	128	256
Ours	0.82	0.88	0.94	1.3	1.9
[33]	0.89	1.2	4.1	*OOM*	*OOM*

out the need of any alignment process. These relationship maps are visualized without the guidance of the blur mask based gating function to get a better intuition into the affinity finding ability of the spatio-temporal attention module. Further, we visualize the ability of the blur mask detector module in Fig. 8 for a few frames. We can observe that for a blurry frame (Fig. 8: left) it is able to detect the blurry regions whereas for a relatively sharp frame (Fig. 8: right) it is successfully identifying the absence of severe blur. The use of blur-mask based gating operation ensures giving more weightage to the regions with relevant information as well as less degradation.

For Net5, we replace the attention module with NLN block from [33]. As shown in Table 5, it requires huge downsampling to implement even a single block, and as a result, losses most of the finer pixel details. Thus, the inclusion of NLN block results in small improvement.

Analysis on the number of selected frames: Simply stacking more neighboring frames in most of the existing works can improve the performance marginally as misalignment between distant frames will be counter productive. In comparison, for our non-local information fusion technique that does not depend on the alignment between frames, we can effectively control the performance or the speed of the network by varying the number of neighboring frames. But, training the network with a fixed number of frames and simply changing it at inference time is suboptimal as the network will not be suited to more/less neighboring frames. Instead, after training the Net4 with a maximum of 5 neighboring frames for a fair comparison with previous SOTA methods, we further fine-tune it by varying the user-defined variable u (which denotes the maximum number of neighboring frames that the user wants to use) and adjusting the reward (Eq. 9) accordingly. With this training strategy, we can vary the network behavior dynamically at inference time. In Table 4, we report the variation of PSNR for a varying number of frames. Also, we have observed that more than 50% of the picked frames are in the temporal range of $\pm(5-10)$. This denotes that limiting the neighborhood to a fixed $\pm 1/\pm 2$ is indeed suboptimal.

6. Conclusion

We have proposed an adaptive approach for video deblurring. We select the frames with the most complementary information from a large neighborhood and then fuse it using a gated spatio-temporal attention module. The proposed model performs favorably against state-of-the-art methods while being efficient. Our approach also allows the user to tune the behavior of the model at inference time to focus on either performance or speed. Such a system can be extended to existing video deblurring methods or other video-processing tasks and will be explored in future.

7. Acknowledgement

Funding from Institute of Eminence (IoE) project No. SB20210832EEMHRD005001 is gratefully acknowledged.

References

[1] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 60–65. IEEE, 2005.

[2] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[3] Huaijin Chen, Jinwei Gu, Orazio Gallo, Ming-Yu Liu, Ashok Veeraraghavan, and Jan Kautz. Reblur2deblur: Deblurring videos via self-supervised learning. In *2018 IEEE International Conference on Computational Photography (ICCP)*, pages 1–9. IEEE, 2018.

[4] Sunghyun Cho, Jue Wang, and Seungyong Lee. Video deblurring for hand-held cameras using patch-based synthesis. *ACM Transactions on Graphics (TOG)*, 31(4):1–9, 2012.

[5] Hehe Fan, Zhongwen Xu, Linchao Zhu, Chenggang Yan, Jianjun Ge, and Yi Yang. Watching a small portion could be as good as watching all: Towards efficient video classification. In *IJCAI International Joint Conference on Artificial Intelligence*, 2018.

[6] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. *arXiv preprint arXiv:1705.08926*, 2017.

[7] Ryosuke Furuta, Naoto Inoue, and Toshihiko Yamasaki. Fully convolutional network with multi-step reinforcement learning for image processing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3598–3605, 2019.

[8] Dong Gong, Jie Yang, Lingqiao Liu, Yanning Zhang, Ian Reid, Chunhua Shen, Anton Van Den Hengel, and Qinfeng Shi. From motion blur to motion flow: a deep learning solution for removing heterogeneous motion blur. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2319–2328, 2017.

[9] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[10] Tae Hyun Kim and Kyoung Mu Lee. Generalized video deblurring for dynamic scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5426–5434, 2015.

[11] Tae Hyun Kim, Kyoung Mu Lee, Bernhard Scholkopf, and Michael Hirsch. Online video deblurring via dynamic temporal blending network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4038–4047, 2017.

[12] Tae Hyun Kim, Mehdi SM Sajjadi, Michael Hirsch, and Bernhard Scholkopf. Spatio-temporal transformer network for video restoration. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 106–122, 2018.

[13] Hailin Jin, Paolo Favaro, and Roberto Cipolla. Visual tracking in the presence of motion blur. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 18–25. IEEE, 2005.

[14] Stamatios Lefkimmiatis. Non-local color image denoising with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3587–3596, 2017.

[15] Xia Li, Zhisheng Zhong, Jianlong Wu, Yibo Yang, Zhouchen Lin, and Hong Liu. Expectation-maximization attention networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9167–9176, 2019.

[16] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Advances in neural information processing systems*, pages 2802–2810, 2016.

[17] Yasuyuki Matsushita, Eyal Ofek, Weina Ge, Xiaoou Tang, and Heung-Yeung Shum. Full-frame video stabilization with motion inpainting. *IEEE Transactions on pattern analysis and Machine Intelligence*, 28(7):1150–1163, 2006.

[18] Christopher Mei and Ian Reid. Modeling and generating complex motion blur for real-time tracking. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[19] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3883–3891, 2017.

[20] Seungjun Nah, Sanghyun Son, and Kyoung Mu Lee. Recurrent neural networks with intra-frame iterations for video deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8102–8111, 2019.

[21] Jinshan Pan, Haoran Bai, and Jinhui Tang. Cascaded deep video deblurring using temporal sharpness prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3043–3051, 2020.

[22] Rui Qian, Robby T Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive generative adversarial network for raindrop removal from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2482–2491, 2018.

[23] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models. *arXiv preprint arXiv:1906.05909*, 2019.

[24] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1279–1288, 2017.

[25] Maitreya Suin, Kuldeep Purohit, and AN Rajagopalan. Degradation aware approach to image restoration using knowledge distillation. *IEEE Journal of Selected Topics in Signal Processing*, 15(2):162–173, 2020.

[26] Maitreya Suin, Kuldeep Purohit, and AN Rajagopalan. Spatially-attentive patch-hierarchical network for adaptive

motion deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3606–3615, 2020.

[27] Maitreya Suin and AN Rajagopalan. An efficient framework for dense video captioning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12039–12046, 2020.

[28] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

[29] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8174–8182, 2018.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[31] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson WH Lau. Spatial attentive single-image deraining with a high quality real rain dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12270–12279, 2019.

[32] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[33] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[34] Patrick Wieschollek, Michael Hirsch, Bernhard Scholkopf, and Hendrik Lensch. Learning blind motion deblurring. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 231–240, 2017.

[35] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Wei Liu, and Hongdong Li. Adversarial spatio-temporal learning for video deblurring. *IEEE Transactions on Image Processing*, 28(1):291–301, 2018.

[36] Zhihang Zhong, Ye Gao, Yinqiang Zheng, and Bo Zheng. Efficient spatio-temporal recurrent neural network for video deblurring.

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