Efficient Video Super Resolution by Gated Local Self Attention

Davide Abati
dabati@qti.qualcomm.com
Amir Ghodrati
ghodrati@qti.qualcomm.com
Amirhossein Habibian
habibian@qti.qualcomm.com

Qualcomm AI Research[1]

[1]Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

Abstract

We tackle the task of efficient video super resolution. Motivated by our study on the quality vs. efficiency trade-off on a wide range of video super resolution architectures, we focus on the design of an efficient temporal alignment module, as it represents the major computational bottleneck in the current solutions. Our alignment module, named Gated Local Self Attention (GLSA), is based on a self-attention formulation and takes advantage of motion priors existing in the video to achieve a high efficiency. More specifically, we leverage the locality of motion in adjacent frames to aggregate information from a local neighborhood only. Moreover, we propose a gating module capable of learning binary functions over pixels, to restrict the alignment only to regions that undergo significant motion. We experimentally show the effectiveness of our proposed alignment on the commonly-used REDS and Vid4 datasets, reducing the overall computational cost by $\sim$13× and $\sim$2.8× respectively compared to state-of-the-art efficient video super-resolution networks.

1 Introduction

Video Super-resolution (VSR) aims at reconstructing a high-resolution (HR) video from its corresponding low-resolution counterpart. The task has drawn much attention recently due to its importance in smart-phone camera use cases, such as zooming, as well as its application in video surveillance and high-definition displays. Differently from single image super-resolution techniques, which only rely on spatial information to reconstruct HR pixels, current state-of-the-art video super-resolution methods aggregate additional temporal information across frames to further enhance details. Particularly, astounding performance in VSR is commonly obtained by aligning pixels in neighboring support frames to the input reference frame, by means of optical flow estimation [30, 37], 3D convolutions [16, 18], or deformable convolutions [33, 36]. While many of these methods have been focused on the benefits of temporal information on reconstruction quality, its impact on computation has not been investigated thoroughly. Through an empirical study on commonly used architectures, we show that even though the alignment is crucial for reconstruction quality, it carries a significant computational overhead in resource-limited scenarios.

A promising approach to eliminate the need for costly explicit alignment is to use self-attention [26, 28, 34]. The attention operation can seamlessly cope with misalignment in sequences by matching an input token to a set of context tokens using a compatibility function, and it is now widely employed for representation learning in language modeling [34], image [14, 21] and video [10, 35] tasks.

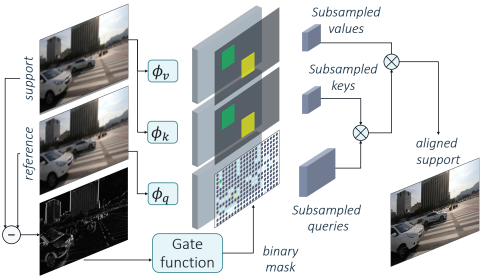


Figure 1: **Gated local self attention (GLSA).** By subsampling of key, query, and value pixels using motion locality and learned gates, GLSA efficiently aligns the information between support and reference frames. Yellow and green colors highlight local supports for two query pixels, and ϕ denotes the embeddings for query, key, and values.

However, the potential of self-attention as a pixel-level alignment operator is still under-explored, with a couple of works on video segmentation [13, 25]. These solutions are not directly applicable to the video super resolution problem, as they are sub-optimal in two aspects: First, they attend globally, by accounting for every pixel in the neighboring support frame, which may lead to overlooking local information that is crucial for an accurate reconstruction. Second, they assume that every pixel has to be aligned, which may lead to inefficiency as many pixels may not undergo significant motion. To address these challenges, we propose a novel attention-based alignment model, named Gated Local Self Attention (GLSA), as shown in Fig. 1, tailored for the task of efficient video super resolution. Our proposal takes advantage of two motion priors existing in the video domain. First, we leverage the locality of motion among neighboring support frames. As pixels typically undergo small displacements between consecutive frames, we restrict the attended context to a local spatial neighborhood, making the alignment operation more efficient. Moreover, not all pixels undergo the same amount of motion. Therefore, alignment is computationally wasteful in regions not affected by any displacement: as such, we limit the alignment operations only on the regions with significant changes. This trait is enabled by designing a gating module capable of learning binary functions over pixels, indicating, conditioned on the current input, where alignment can be skipped. The module is fed with the residual between support and reference frames, and the binary gates are trained jointly with the super resolution model, by means of the Gumbel-Softmax reparametrization. Our alignment model, set up within a simple CNN backbone network, achieves $\sim$13× and $\sim$2.8× reduction in computational cost on the REDS and Vid4 datasets respectively, compared to the most efficient VSR models. We summarize our contributions as follows: *i.* We conduct a systematic study on the quality-efficiency trade-off of current temporal aggregation techniques in VSR, highlighting the central role of the alignment operator. *ii.* We introduce a feature alignment operator based on self-attention, that efficiently aggregates information from a local vicinity in neighboring frames to reconstruct high-resolution details. *iii.* We propose a conditional gating function to restrict the alignment operation to regions that undergo significant motion across frames. This further improves the efficiency as the alignment module learns to skip unnecessary computation for stationary pixels.

2 Related Work

Video Super Resolution. Current VSR models can be categorized in two aspects: *i.* architectural designs for temporal integration and *ii.* operations used for aligning frames.
In terms of architecture, one solution is the use of *multi-frame* designs [11, 16, 18, 20, 33, 36], which feed the model with multiple support frames and exploit them in several ways: for example, by explicit alignment and fusion [20, 33, 36], or by 3D convolutions [16, 18]. On the other hand, *recurrent* architectures restrict the support information to the previous timestep only [15, 30] or use bidirectional schemes [5], and are typically more efficient than multi-frame models. As we focus on efficiency, we deploy our alignment operation in recurrent architectures. However, our method can be in principle applied to both families.
In terms of alignment operators, early models mainly rely on optical flow [3, 30, 32, 37]. More recently, designs based on deformable convolutions [7, 40] gained popularity, spanning from single layers [33] to feature pyramids [36]. Another strategy is implicit alignment, by processing frames with 3D [16, 18] or 2D [9, 15] convolutions. Unlike previous models, our work explores self-attention as an alignment operator.

Self attention. Since its original formulation for language modeling [34], self-attention has found an increasing use in vision applications, such as action recognition [10, 35], object detection [4], segmentation [13, 25] and classification [8, 28, 39]. A lot of works have also been directed towards efficient self-attention, by sparsifying the attention matrix with block-wise [27], local [26, 28] or sparse hand-crafted patterns [1, 6, 12]. In this work, we rely on local self-attention for efficiency, however restricting its application to a subset of queries, conditioned on the current input, to save more computation.
A few models employ attention mechanisms for image [21] and video [16, 38] super resolution. However, in these efforts attention represents a mean to improve or suppress representations rather than temporally aligning them. As such, these methods rely on coarse schemes such as attending entire feature maps [16] or spatio-temporal tubelets [38]. In contrast, we employ self-attention at a fine pixel-level scale, for solving misalignments.

3 Temporal Modeling in VSR Architectures

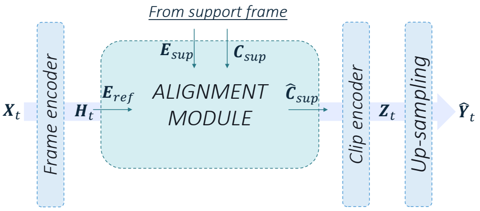


Figure 2: Overview of VSR architectures.

VSR aims at reconstructing a sequence of high-resolution frames $\mathcal{Y} = [\mathbf{Y}_1, \dots, \mathbf{Y}_T]$, with $\mathbf{Y}_i \in \mathbb{R}^{c \times sh \times sw}$, from their low-resolution counterparts, denoted by $\mathcal{X} = [\mathbf{X}_1, \dots, \mathbf{X}_T]$, where each $\mathbf{X}_i \in \mathbb{R}^{c \times h \times w}$ and s represents the targeted upsampling factor. The typical way to obtain the low-resolution input clip $\mathcal{X}$ is by applying a specific downsampling kernel to each $\mathbf{Y}_t$, characterizing VSR as a complicated inverse problem.

VSR architectures. Many VSR architectures can be described, at a high level, as follows (see Fig. 2). The input frame $\mathbf{X}_t$ goes through a frame encoder defined as a cascade of residual blocks. The encoded frame $\mathbf{H}_t$ is then fed to an alignment function, along with feature maps from one or multiple support frames. The alignment function involves motion estimation and compensation steps to spatially align the features from one or more support frames to the reference frame. We formalize the alignment function as:

$$\hat{\mathbf{C}}_{sup} = \mathrm{A}(\mathbf{E}_{ref}, \mathbf{E}_{sup}, \mathbf{C}_{sup}), \tag{1}$$

where $\mathbf{E}_{ref}$ and $\mathbf{E}_{sup}$ denote the feature maps from the reference and support frame respectively, that is used for motion estimation. $\mathbf{C}_{sup}$ denotes the support feature map to be aligned based on the estimated motion, and $\hat{\mathbf{C}}_{sup}$ is the aligned features to be passed to the clip encoder. Clip encoder feeds the aligned features, together with the frame encoding $\mathbf{H}_t$, to a cascade of residual blocks to generate a clip encoding $\mathbf{Z}_t$. Finally, the high resolution estimate $\hat{\mathbf{Y}}_t$ is obtained by upsampling the clip encoding using sub-pixel convolution [31] and by summing it with the bicubic magnification of the frame, as in [36].

Quality vs. Efficiency: an empirical study. Many existing VSR architectures can be instantiated from the aforementioned formulation based on how to choose the alignment inputs. To understand the impact of alignment on the quality and efficiency of VSR models, we conduct an empirical study on a wide range of architectures. Specifically, we fix $\mathbf{E}_{ref} = \mathbf{H}_t$ and consider:

- **MF** [16, 20, 36]: a multi-frame architecture that aligns a set of $K = 7$ neighboring frames as $\mathbf{E}_{sup} = \mathbf{C}_{sup} := \{\mathbf{H}_{t-\frac{K-1}{2}}, \ldots, \mathbf{H}_{t+\frac{K-1}{2}}\}$
- **REC-H** [15]: a recurrent architecture that aligns the clip encoding as $\mathbf{E}_{sup} := \{\mathbf{H}_{t-1}\}$ and $\mathbf{C}_{sup} := \{\mathbf{Z}_{t-1}\}$.
- **REC-Y** [30]: a recurrent architecture that aligns the previous output as $\mathbf{E}_{sup} := \{\mathbf{H}_{t-1}\}$ and $\mathbf{C}_{sup} := \{\hat{\mathbf{Y}}_{t-1}\}$.
- **REC-H+REC-Y**: a recurrent architecture that aligns the both clip encoding and output as $\mathbf{E}_{sup} := \{\mathbf{H}_{t-1}\}$ and $\mathbf{C}_{sup} := \{\mathbf{Z}_t - 1, \hat{\mathbf{Y}}_{t-1}\}$.
- **MF+REC-H**: a multi-frame architecture with a clip encoding feedback as $\mathbf{E}_{sup} := \{\mathbf{H}_{t-\frac{K-1}{2}}, \ldots, \mathbf{H}_{t+\frac{K-1}{2}}\}$ and $\mathbf{C}_{sup} := \{\mathbf{H}_{t-\frac{K-1}{2}}, \ldots, \mathbf{H}_{t+\frac{K-1}{2}}, \mathbf{Z}_{t-1}\}$.
- **MF+REC-Y**: a multi-frame architecture with an output feedback as $\mathbf{E}_{sup} := \{\mathbf{H}_{t-\frac{K-1}{2}}, \ldots, \mathbf{H}_{t+\frac{K-1}{2}}\}$ and $\mathbf{C}_{sup} := \{\mathbf{H}_{t-\frac{K-1}{2}}, \ldots, \mathbf{H}_{t+\frac{K-1}{2}}, \hat{\mathbf{Y}}_{t-1}\}$.
- **MF+REC-H+REC-Y** [9]: a multi-frame architecture with feedback on both clip encoding and output as $\mathbf{E}_{sup} := \{\mathbf{H}_{t-\frac{K-1}{2}}, \ldots, \mathbf{H}_{t+\frac{K-1}{2}}\}$ and $\mathbf{C}_{sup} := \{\mathbf{H}_{t-\frac{K-1}{2}}, \ldots, \mathbf{H}_{t+\frac{K-1}{2}}, \mathbf{Z}_{t-1}, \hat{\mathbf{Y}}_{t-1}\}$.

For each architecture, we consider models both in the absence and the presence of an alignment operation. In the latter case, we rely on a deformable convolution [7, 40], which is the current state-of-the-art for VSR [33, 36].

We conduct our study on the REDS dataset [24] using a light backbone[1]. We summarize our results in Fig. 3, where we report, for every model, the PSNR gain and the computational overhead with respect to an image-based counterpart with no temporal modeling. The results show that, under all architectural designs, the use of support information is only beneficial in the presence of the alignment operator: indeed, the alignment module allows for a significant increase in PSNR, spanning from 0.52 db for recurrent models up to 0.67 db to more complex

[1] 2 and 5 residual blocks, with 32 channels, as frame and clip encoder

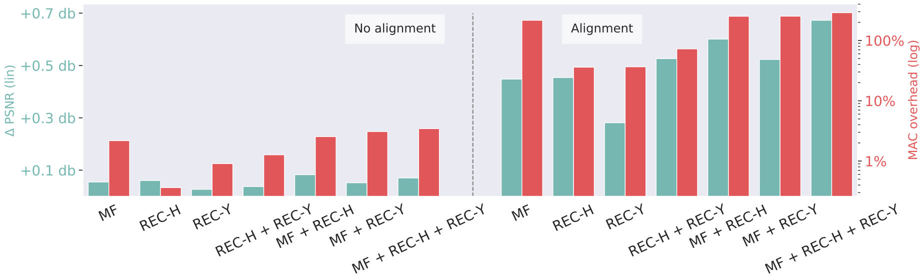


Figure 3: **Quality vs. Efficiency in VSR architectures.** Green and red bars represent the increase in quality (PSNR) and computation (MAC). **(Left)** Without alignment, temporal information from support frames is marginally beneficial w.r.t an image model (28.26 db / 16.55 GMACs). **(right)** The same architectures prove successful whenever alignment is applied. However, the improvement comes with significant computational overhead.

designs. On the contrary, support information, under any architectural choice, in the absence of alignment can hardly improve the image-based model, with gains always under 0.1 db. However, alignment improvements also come at a significant cost. Indeed, the overhead of alignment is at least 36% in the case of recurrent models, and can increase up to 288% when multi-frame schemes are involved. The finding suggests the need for efficient alignment solutions that can enable VSR to operate in low-computational, high-accuracy regimes.

4 Gated Local Self Attention

Self-attention as an alignment operator. Following the notations introduced in Sec. 3, we associate $\mathbf{E}_{ref}$, $\mathbf{E}_{sup}$ and $\mathbf{C}_{sup}$ to queries, keys and values embeddings used in self-attention:

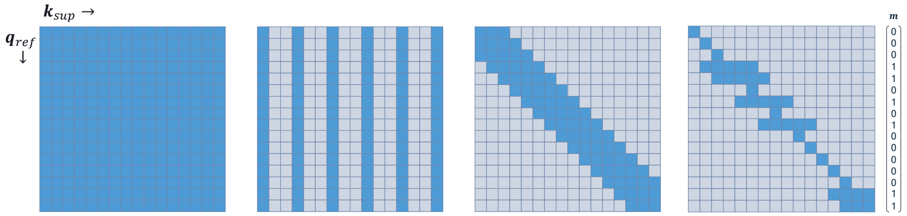
$$\mathbf{Q}_{ref} = \phi_q(\mathbf{E}_{ref}) \in \mathbb{R}^{hw \times d_k} \qquad \mathbf{K}_{sup} = \phi_k(\mathbf{E}_{sup}) \in \mathbb{R}^{hw \times d_k} \qquad \mathbf{V}_{sup} = \phi_v(\mathbf{C}_{sup}) \in \mathbb{R}^{hw \times d_v}, \tag{2}$$

where each ϕ is a pixel-wise projection function, parametrized separately for q, k and v, followed by reshaping, and d_k and d_v represent the dimensionality of keys and values. In this formulation, each reference pixel in $\mathbf{E}_{ref}$ is represented by a query vector. Moreover, support pixels in $\mathbf{E}_{sup}$ and $\mathbf{C}_{sup}$ are represented by key and value vectors, defining a $h \times w$ search space. The alignment operator is then described as

$$\mathbf{Att} = \mathrm{softmax}(\mathbf{Q}_{ref}\mathbf{K}_{sup}^T), \quad \text{(3a)} \qquad\qquad \hat{\mathbf{C}}_{sup} = \mathbf{Att} \cdot \mathbf{V}_{sup}, \quad \text{(3b)}$$

where $\mathbf{Att}$ is a dense $hw \times hw$ matrix holding normalized pairwise similarities between all the reference queries and support keys. Eq. 3a resembles the motion estimation step in an alignment module, as the attention matrix $\mathbf{Att}$ can inform about the spatial displacement between the reference and support features. Intuitively, in this stage, each query explores the key search space to find the most similar ones to itself. Next, Eq. 3b computes the output aligned feature map $\hat{\mathbf{C}}_{sup} \in \mathbb{R}^{hw \times d_v}$: this operation resembles the motion compensation step, as it transforms the support values, $\mathbf{V}_{sup}$, according to the attention matrix.

In summary, self-attention gracefully fits the requirements for its use as an alignment module. However, the current formulation is sub-optimal in terms of computation. Next, we propose our key and query subsampling strategies, making it suitable for efficient VSR.



(a) Global (b) Key-downsample (c) Local (d) GLSA

Figure 4: **Impact of key and query subsampling on the sparsity of the attention matrix.** (a) In global self-attention, every query is compared to all the the keys. (b) Keys are subsampled to shrink the search space as [13]. (c) A local self-attention searches in a local $k = 2$ neighborhood of each query. (d) our proposed method subsamples both keys and queries.

Local key subsampling. The global search space defined by Eq. 2 and illustrated in Fig. 4 (a), carries a quadratic cost in the number of pixels which makes global self-attention often prohibitive to compute. A popular solution is to spatially subsample the key search space [13, 35] as shown in Fig. 4 (b). However, this solution is still based on a global search that is sub-optimal as VSR problem concerns the local pixel reconstruction. Moreover, grid-subsampling decreases the spatial resolution and exposes coarser information that is hardly beneficial for the task of super resolution.

Differently, we limit the search in a local $k \times k$ neighborhood of each query, as shown in Fig. 4 (c), resulting in the following embedding dimensionalities:

$$\mathbf{Q}_{ref} \in \mathbb{R}^{hw \times d_k}, \ \mathbf{K}_{sup} \in \mathbb{R}^{k^2 \times d_k}, \ \mathbf{V}_{sup} \in \mathbb{R}^{k^2 \times d_v}. \tag{4}$$

By assuming that motion is limited in the consecutive frames, we drop global search in favor of local search. This allows the search to operate at a fine resolution which is both efficient and desirable for video super resolution task.

Dynamic query subsampling. The locality over keys presented above enables efficiency by limiting the search space for each query. However, in many cases a significant amount of pixels undergoes a negligible motion between consecutive frames and, as such, doesn't require alignment (*e.g.* no camera motion). In this case, we can save computation by carrying out the alignment for moving regions only and skipping it wherever not needed. To this end, we introduce a query subsampling method that learns to adaptively skip alignment in several locations of the reference feature map. In particular, we introduce a binary gating function g, as represented in Fig. 1, that takes as input $\mathbf{E}_{ref}$ and $\mathbf{E}_{sup}$ and computes a binary mask $\mathbf{m} \in \{0,1\}^{hw}$ over pixels, representing for which of them alignment is needed:

$$\mathbf{m} = g(\mathbf{E}_{ref}, \mathbf{E}_{sup}) \begin{cases} \sim \mathrm{Bern}(p) & \text{if training,} \\ = \mathrm{round}(p) & \text{if inference,} \end{cases} \quad \text{where} \quad p = \sigma(f_\theta(\|\mathbf{E}_{ref} - \mathbf{E}_{sup}\|_1)). \tag{5}$$

First, the function g computes the residual feature map between $\mathbf{E}_{ref}$ and $\mathbf{E}_{sup}$, which conveys strong prior cues about which pixels are affected by motion. Then, the residual is fed to a single-channel convolution f_θ, parametrized by θ, followed by sigmoid activation function σ. The result, $p \in [0,1]^{hw}$, represents, for each pixel, a soft decision on performing or skipping alignment. In order to train hard decisions, we rely on the Gumbel-Softmax

reparametrization [17, 22] to sample stochastic Bernoulli realizations of the mask $\mathbf{m}$. In the forward pass, we add noise to the output of f_θ, sampled from a Gumbel distribution, before applying the sigmoid and rounding to $\{0,1\}$. In the backward pass, we get a biased estimate of the gradient by employing the straight-through estimator [2]: as such, we bypass the rounding operation and backpropagate the gradient as-is. During inference, the binary mask is obtained simply by rounding the sigmoid-activated values p.
The gating module g enables a formalization of the alignment operator whose computational complexity varies depending on the input content, by modulating the number of active queries through the mask $\mathbf{m}$, resulting in the following embedding dimensionalities:

$$\mathbf{Q}_{ref} \in \mathbb{R}^{\|\mathbf{m}\|_1 \times d_k},\ \mathbf{K}_{sup} \in \mathbb{R}^{k^2 \times d_k},\ \mathbf{V}_{sup} \in \mathbb{R}^{k^2 \times d_v} \tag{6}$$

where $\|\mathbf{m}\|_1 \ll h \times w$ represents the number of active queries. Specifically, wherever $\mathbf{m} = 1$, pixels undergo alignment as specified in Eq. 3a, 3b. On the contrary, where $\mathbf{m} = 0$, alignment is skipped and $\hat{\mathbf{C}}_{sup} = \mathbf{C}_{sup}$. We can understand the latter case as sparse rows in the attention matrix, filled with zeros except for the main diagonal, as depicted in Fig. 4 (d).
When training the model, gates may naturally learn to activate all queries, as the super resolution objective only aims for maximizing the quality of the reconstruction. Therefore, they may not bring any save in computational efficiency. To overcome this problem, we regularize the gates by enforcing them to fire only where alignment is strictly needed. We thus introduce an sparsity objective as an l_1 regularization over gates: $\mathcal{L}_{gate} = \|\mathbf{m}\|_1$. The gating parameters θ are learned jointly with the model parameters by minimizing the overall objective $L_{vsr} + \beta L_{gate}$. The hyper-parameter β balances the model accuracy, measured by L_{vsr}, and the alignment efficiency, measured by L_{gate}.

5 Experiments

We start by reporting ablation studies on query subsampling. Then, we fix the backbone model and compare our proposal against current alignment solutions in terms of accuracy and efficiency. Finally, we report a comparison with several state-of-the-art VSR models.

Datasets and metrics. We conduct experiments on two datasets: First, we employ the REDS dataset [23], and follow the partitions defined in [36], comprising 266 clips for training and 4 clips for testing (REDS4). Each clip contains 100 frames. As a second dataset, we train our models on 83,877 Vimeo90k [37] 7-frame clips, and test them on the commonly used Vid4 benchmark. In both datasets, we tackle 4x upsampling. For REDS, we use as input the released low-resolution sequences, downgraded with bicubic interpolation (BI). For Vid4 we rely on gaussian downsampling (BD) [18] as used in [9, 18, 30, 33]. We rely on Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM), to assess reconstruction quality. For every reported result on REDS4, PSNR and SSIM metrics are computed on RGB channels. For Vid4, metrics are computed on the Y channel. We measure the computational cost as Multiply-Accumulate Count (MAC) required per-frame. For our model, we include the small overhead induced by gating modules in the MAC computation. We plug the GLSA alignment module into a simple convolutional backbone as depicted in Fig. 2, of which we define two variants at two different MAC operating points. The lighter backbone (B0) is composed of 2 and 5 residual blocks, each with 32 channels, as frame and clip encoders respectively. In the heavier backbone (B1) frame and clip encoders are made of 5 and 10 residual blocks with 64 channels. In all the experiments, we rely on a

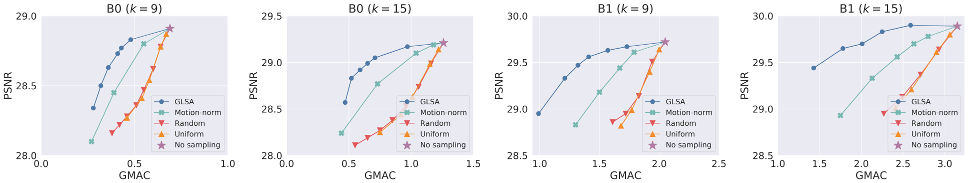


Figure 5: **Dynamic query subsampling.** Our proposed gating function outperforms other sampling baselines for different backbones and attention kernel sizes k.

recurrent architecture (**REC-H**, see Sec. 3) since it yields the best tradeoff between accuracy and efficiency.

Optimization. Our models are trained on 7-frames random clips using Adam [19] optimizer for 100 epochs using batches of 16 clips. On REDS dataset, we optimize a Charbonnier objective with an initial learning rate of 0.0004. On Vimeo90k dataset, we optimize the mean squared error with an initial learning rate set to 0.0001. We decay the learning rate by a factor of 10 after 60 epochs. Unless otherwise specified, we fix the local neighborhood for key subsampling to $k = 21$. Gate parameters are learned by fine-tuning the model using the same optimizer with 10× lower learning rate. To get different accuracy vs. efficiency tradeoffs, we vary the regularization parameter β within $[50, 300]$ for REDS and within $[1, 10]$ for the Vimeo90k dataset. Precise configurations of β can be found in the supplement.

Dynamic query subsampling. We compare the proposed dynamic query subsampling to four baselines: *i.* No sampling, meaning local self-attention without query selection; *ii.* Random sampling, that limits the query to a random set of pixels; *iii.* Uniform sampling, that selects queries on a strided grid; *iv.* Motion-norm, that similar to GLSA samples the query pixels based on their motion. However, instead of using the learnable gating function it simply relies on the magnitude of feature differences to select query pixels. More specifically, it limits the query to pixels for which $||\mathbf{E}_{ref} - \mathbf{E}_{sup}||_1$ exceeds a preset threshold. Fig. 5 reports the results for two backbones using two different k: 9×9 and 15×15. GLSA and Motion-norm, which rely on motion information for sampling, consistently outperform Uniform and Random samplings. This finding advocates for the intuition that motion provides a strong cue on what pixels greatly benefit from the alignment. Moreover, the superior performance of GLSA compared to Motion-norm suggests that the motion alone is not sufficient for an effective query subsampling.

Alignment methods comparison. We now provide a fair assessment of different alignment operators, by comparing them in the same settings. For every alternative we use the B0 backbone with **REC-H** architecture and the same training pipeline. We carry out the experiment on REDS, and consider the following alignment modules: *i)* None, an image-based baseline; *ii)* optical-flow based alignment, using either SpyNet [29] as done in [5, 37] or the shallow network used in [3, 32] (OF-simple); *iii)* deformable convolutions, either as a single operator [7, 40] (dconv), or in the pyramidal scheme in [36] (PCD); *iv)* the correlation-based search in MuCAN [20].

	REDS4					
	000	011	015	020	avg	GMAC
Bicubic	24.63 / 0.6529	26.17 / 0.7290	28.61 / 0.8042	25.52 / 0.7417	26.24 / 0.7319	-
ToFlow [37]	26.52 / 0.7540	27.80 / 0.7858	30.67 / 0.8609	26.92 / 0.7953	27.98 / 0.7990	133.01
DUF 52L [18] †	27.30 / 0.7937	28.38 / 0.8057	31.55 / 0.8847	27.30 / 0.8165	28.64 / 0.8251	1662.00
EDVR-M [36]	27.75 / 0.8153	31.29 / 0.8732	33.48 / 0.9133	29.59 / 0.8776	30.53 / 0.8698	463.50
EDVR-S	27.48 / 0.8048	30.71 / 0.8628	32.92 / 0.9061	29.16 / 0.8682	30.07 / 0.8605	300.47
EDVR-XS	27.26 / 0.7941	30.18 / 0.8525	32.55 / 0.9002	28.83 / 0.8598	29.70 / 0.8517	157.2
EDVR-XXS	27.05 / 0.7846	29.58 / 0.8382	32.02 / 0.8911	28.35 / 0.8466	29.25 / 0.8401	81.01
RLSP 7-64 [9]	26.21 / 0.7367	28.75 / 0.8114	31.31 / 0.8740	27.43 / 0.8169	28.43 / 0.8097	16.28
RLSP 7-256 [9]	27.25 / 0.7941	29.56 / 0.8319	32.38 / 0.8965	28.02 / 0.8363	29.30 / 0.8397	243.47
GLSA-B0	26.88 / 0.7775	29.72 / 0.8400	31.93 / 0.8892	28.43 / 0.8483	29.24 / 0.8388	18.61
GLSA-B1	27.36 / 0.8003	30.60 / 0.8595	32.73 / 0.9024	29.03 / 0.8644	29.93 / 0.8566	88.85

Table 1: **State of the art on REDS4.** Our GLSA model is computationally cheap and outperforms existing approaches. Performances are reported as (PSNR / SSIM) on RGB channels. GMAC computed for an input size of 180×320. † Results based on BD downsampling.

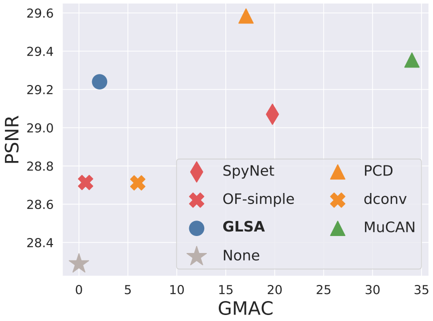


Figure 6: **PSNR/MAC trade-off of alignment operators.** For all the alignment methods we use the same backbone (B0, REC-H) and training pipeline. GLSA achieves the best tradeoff with only ∼13% computational overhead with respect to the image-based model. GMACs are hereby reported for the alignment operator only (i.e. backbone is not accounted).

The results are reported in the MAC/PSNR plot in Fig. 6. In terms of PSNR the best alignment is obtained by the PCD module, that however introduces a significant computational overhead. The optical flow alignment based on SpyNet is comparable in computational cost to the PCD alignment, but shows a 0.51 db drop in PSNR. The efficiency can be increased both for optical flow and deformable convolution based alignment by using simpler modules. However, this strategy incurs a significant PSNR reduction, as testified by the OF-simple and dconv models. Our GLSA operator for alignment achieves the best trade-off, being able to improve the performance over the baseline image model (16.55 GMAC) by ∼1 db with only a 13% MAC overhead.

Comparison to state of the art. On the REDS dataset, we compare with RLSP [9], a recurrent architecture designed for efficiency, that we trained it by using the code publicly released by the authors. As shown in Table 1, our B0 significantly outperforms the 7-64 architecture, at a comparable computational cost. Moreover, with similar reconstruction quality, B0 requires ∼13x less computation than 7-256. Finally, our best model, B1, improves over RSLP 7-256 by 0.63 db, while being ∼2.7 times more efficient. We additionally compare GLSA to standard multi-frame models that operate in a high computational regime, namely TOFlow [37], DUF [18] and EDVR-M [36], relying on optical-flow, 3D convolutions, and deformable convolutions for alignment respectively. To enable a fair comparison at comparable GMACs, we also train cheaper EDVR models (S,XS,XXS) with the code released by the authors. The details of such models, streamlined from EDVR-M by reducing the number of residual blocks and their filters, are presented in the supplementary material.

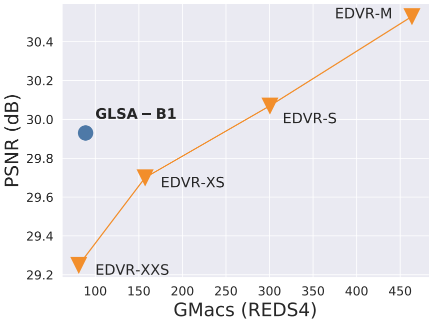


Figure 7: **Comparison with EDVR cheap variants.** GLSA achieves a better PSNR under the same computational budget.

Our B1 model outperforms ToFlow and DUF by a large margin (more than 1 db in PSNR) in reconstruction quality while using lower MAC overall. It's also noteworthy that even our B0 model still outperforms them with ∼7x and ∼90x less computation, respectively. On the quality side, EDVR-M performs the best among all competing methods, but its cost is significantly higher with respect to our method. It is interesting compare our proposal with cheap EDVR models, for which we report PSNR / GMACs plots in Fig. 7. GLSA achieves achieves similar performances w.r.t. EDVR-S, requiring 29.57% of the GMACs. Comparing to EDVR variant with closest cost, EDVR-XXS, our proposal outperforms its PSNR by a 0.7 db. Some reconstruction examples are reported in the supplementary document.

Similar trends can be observed for Vid4, for which the comparison is reported in Table 2. As the table shows, GLSA performs on par with RLSP 7-256 in terms of reconstruction quality, while being ∼2.8x more efficient. Our proposal also outperforms expensive models, except for EDVR-M, while requiring less computation. It is interesting to notice how the gap between RLSP and alignment based methods, such as GLSA and EDVR-M, is reduced on Vid4 as compared to REDS4. We conjecture that the reason of this behavior lies in the higher motion that characterizes REDS4, comprising hand-held fast camera movements. In those settings, the absence of an explicit alignment module significantly impacts the reconstruction quality, contrarily to the case of Vid4, whose mild moving patterns that can be modeled implicitly.

6 Conclusions

In this paper, we tackled efficient video super resolution. We showed that although alignment is a crucial step to obtain high reconstruction quality, it comes at a significant computational cost. As a consequence, we presented GLSA, an efficient alignment method based on local self-attention that operates in low-computational, high-accuracy regimes. To achieve efficiency, our proposed method uses a gating function, trained to allow the network to adaptively skip alignment operations on regions in the frame with negligible motion. Comprehensive experiments show the suitability of our model in balancing accuracy and efficiency in video super resolution.

	PSNR	SSIM	GMAC
SPMC [32]	26.05	0.776	255.22
TDAN [33]	26.58	0.801	126.79
DUF 28L [18]	26.99	0.822	184.80
EDVR-M [36]	27.45	0.841	203.90
FSRVSR [30]	26.69	0.822	80.49
RLSP 7-128 [9]*	26.85	0.821	27.40
RLSP 7-256 [9]*	27.05	0.831	107.12
GLSA-B1	27.04	0.824	37.87

Table 2: **Comparisons with state of the art on Vid4.** PSNR is reported on Y channel. GLSA obtains competitive performance with less computation. GMAC computed for an input size of 144×176. * Results obtained by training with the code publicly released.

References

[1] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[2] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[3] Jose Caballero, Christian Ledig, Andrew Aitken, Alejandro Acosta, Johannes Totz, Zehan Wang, and Wenzhe Shi. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *CVPR*, 2017.

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[5] Kelvin CK Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Basicvsr: The search for essential components in video super-resolution and beyond. *CVPR*, 2021.

[6] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

[7] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *CVPR*, 2017.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

[9] Dario Fuoli, Shuhang Gu, and Radu Timofte. Efficient video super-resolution through recurrent latent space propagation. In *ICCVW*, 2019.

[10] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *CVPR*, 2019.

[11] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Recurrent back-projection network for video super-resolution. In *CVPR*, 2019.

[12] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019.

[13] Ping Hu, Fabian Caba, Oliver Wang, Zhe Lin, Stan Sclaroff, and Federico Perazzi. Temporally distributed networks for fast video semantic segmentation. In *CVPR*, 2020.

[14] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *CVPR*, 2019.

[15] Takashi Isobe, Xu Jia, Shuhang Gu, Songjiang Li, Shengjin Wang, and Qi Tian. Video super-resolution with recurrent structure-detail network. *ECCV*, 2020.

[16] Takashi Isobe, Songjiang Li, Xu Jia, Shanxin Yuan, Gregory Slabaugh, Chunjing Xu, Ya-Li Li, Shengjin Wang, and Qi Tian. Video super-resolution with temporal group attention. In *CVPR*, 2020.

[17] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *ICLR*, 2017.

[18] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *CVPR*, 2018.

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2014.

[20] Wenbo Li, Xin Tao, Taian Guo, Lu Qi, Jiangbo Lu, and Jiaya Jia. Mucan: Multi-correspondence aggregation network for video super-resolution. *ECCV*, 2020.

[21] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. Non-local recurrent network for image restoration. *NeurIPS*, 2018.

[22] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *ICLR*, 2017.

[23] Seungjun Nah, Radu Timofte, Shuhang Gu, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, and Kyoung Mu Lee. Ntire 2019 challenge on video super-resolution: Methods and results. In *CVPRW*, 2019.

[24] Seungjun Nah, Radu Timofte, Shuhang Gu, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, and Kyoung Mu Lee. Ntire 2019 challenge on video super-resolution: Methods and results. In *CVPRW*, 2019.

[25] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019.

[26] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *ICML*, 2018.

[27] Jiezhong Qiu, Hao Ma, Omer Levy, Scott Wen-tau Yih, Sinong Wang, and Jie Tang. Blockwise self-attention for long document understanding. *EMNLP*, 2020.

[28] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models. *NeurIPS*, 2019.

[29] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *CVPR*, 2017.

[30] Mehdi SM Sajjadi, Raviteja Vemulapalli, and Matthew Brown. Frame-recurrent video super-resolution. In *CVPR*, 2018.

[31] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, 2016.

[32] Xin Tao, Hongyun Gao, Renjie Liao, Jue Wang, and Jiaya Jia. Detail-revealing deep video super-resolution. In *ICCV*, 2017.

[33] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. Tdan: Temporally-deformable alignment network for video super-resolution. In *CVPR*, 2020.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017.

[35] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[36] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *CVPRW*, 2019.

[37] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *IJCV*, 2019.

[38] Peng Yi, Zhongyuan Wang, Kui Jiang, Junjun Jiang, and Jiayi Ma. Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations. In *ICCV*, 2019.

[39] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *CVPR*, 2020.

[40] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *CVPR*, 2019.