

Efficient Video Super Resolution by Gated Local Self Attention - Supplementary Material

Davide Abati

dabati@qti.qualcomm.com

Amir Ghodrati

ghodrati@qti.qualcomm.com

Amirhossein Habibian

habibian@qti.qualcomm.com

Qualcomm AI Research¹

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

1 Experimental details

In this section, we provide more details about the experiments in Sec. 5 of the main paper.

Backbone architecture. We plug our GLSA alignment operator in a backbone model as represented in Fig. 2 in the main paper. More specifically, the architecture is composed of several blocks:

- **Frame encoder:** it is composed of a stem convolution projecting the input frame from 3 to c output channels, followed by a sequence of B_f basic residual blocks [14] without batch-normalization. Every pair of convolutions in residual blocks has c channels.
- **Clip encoder:** operating after the alignment operator, takes as input the reference features \mathbf{e}_{ref} and the aligned support features $\hat{\mathbf{e}}_{sup}$. The two tensors get concatenated and projected to c with a pixelwise convolution. Then, a further sequence of B_c residual blocks is applied to the fused features. As in the frame encoder, every convolutional layer features c channels.
- **Up-sampling:** takes as input the output of the clip encoder and performs a $4\times$ up-sampling with two $2\times$ subpixel convolutions (Depth2Space). A final 3×3 regresses the rgb residual, that is summed to the bilinear upsampling of the reference frame to provide the super resolution estimate, as in [9].

As mentioned in the main paper, we define two instances of the backbone architecture, at two different MAC operating points. The lighter backbone (B0) is composed of $B_f = 2$ and $B_c = 5$ residual blocks, each with $c = 32$ channels, for the frame and clip encoders respectively. In the heavier backbone (B1) frame and clip encoders are made of $B_f = 5$ and $B_c = 10$ residual blocks with $c = 64$ channels. In all the experiments, we rely on a recurrent architecture (REC-H, as described in Sec.3) since it yields the best tradeoff between accuracy and

efficiency. For self-attention, we always use 2 heads, and we fix $d_k = d_v$ to 32 and 64 for B0 and B1 respectively.

Cheaper EDVR variants. In Sec. 5 of the main paper, we enable a fair comparison between GLSA and EDVR by comparing at the same GMACs. To this end, we design cheaper versions of the EDVR model and train them using the publicly released codebase. Details about hyperparameters and architectures are reported in Tab. 1.

Sparsity parameter β . Tab. 2 holds all the configurations we employed for our sparsity hyperparameter β .

2 Local key subsampling

We study the effectiveness of the local key subsampling compared to two key sampling baselines, on the REDS dataset. Both baselines sample keys from the whole spatial extent of the support frame, and comprise: *i*. Random sampling, that reduces the attention keys to a set of randomly selected pixel indices; *ii*. Uniform sampling, that selects keys on a strided grid. Uniform sampling resembles the scheme in [9]. All methods sample the same number of keys ranging from 21×21 , 15×15 , 9×9 and 7×7 . As shown in Fig. 1, local key subsampling consistently outperforms baselines on both backbones. The superiority of local key subsampling (0.7–1.1 db) verifies our assumption that most of the details required for an effective reconstruction of a pixel reside within a local neighborhood over adjacent frames.

3 Effect of kernel size

We study the effect of the kernel size on reconstruction quality and cost. Intuitively, higher kernel sizes translates into better reconstruction but also more computation. Tab. 3 reports the PSNR and GMACs tradeoff on REDS4 of our B1 backbone at different kernel sizes. In the main paper, we use $k = 21$ unless otherwise specified.

4 Gates visualizations

We provide in Fig 2 some examples of query selection in GLSA. We represent the overlay between support and reference frames, as well as the regions selected for alignment under different sparsity constraints. The parameter β controls the amount of pixels undergoing the

	feat. blocks	recon. blocks	filters	GMAC	PSNR
EDVR-L	5	40	128	2047	31.09
EDVR-M	5	10	64	463.5	30.53
EDVR-S	5	5	48	300.47	30.07
EDVR-XS	2	5	32	157.2	29.70
EDVR-XXS	2	5	16	81.01	29.25

Table 1: Hyperparameters for cheaper EDVR variants utilized in Sec. 5 in the main paper.

	Experiment	β
Fig. 5	dynamic query subsampling	$\beta = 50$ to 350
Fig. 8	PSNR/MAC trade-off of alignment operators	$\beta = 50$
Tab. 1	state-of-the-art (REDS4)	$\beta = 50$
Tab. 2	state-of-the-art (Vid4)	$\beta = 1$

Table 2: β **configurations** used in Sec. 5 of the main paper.

	$k = 3$	$k = 5$	$k = 7$	$k = 9$	$k = 15$	$k = 21$
PSNR	29.16	29.36	29.55	29.71	29.89	29.97
GMACs	1.75	1.88	2.06	2.31	3.42	5.80

Table 3: Impact of attention kernel size on PSNR and cost for B1 backbone (no dynamic query subsampling is used). GMACs refer to alignment module only.

alignment phase, trading off accurate models ($\beta = 100$, most pixels are aligned) and efficient models ($\beta = 400$, very few pixels are aligned).

5 Qualitative results

We represent in Fig. 3 some results of GLSA on REDS4, compared to Bicubic, DUF, EDVR-M and RLSP variants.

6 Alignment visualization

In this section, we provide qualitative illustrations of the behavior of the GLSA alignment module. To this end, we rely on models trained for video super resolution, and represent the effect of warping on input frames. Specifically, we rely on our B0 backbone model, equipped with GLSA with local key subsampling in a 9×9 neighborhood (*i.e.* $k = 9$). Although

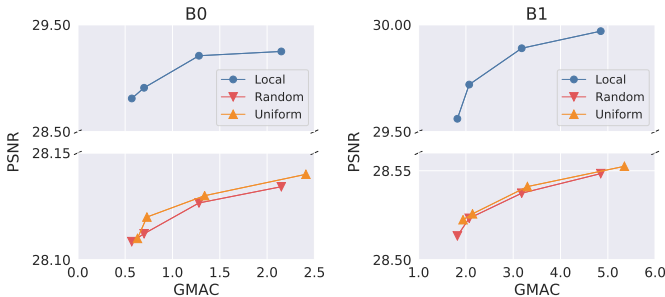


Figure 1: **Local key subsampling** consistently outperforms random and uniform baselines on the both backbones.

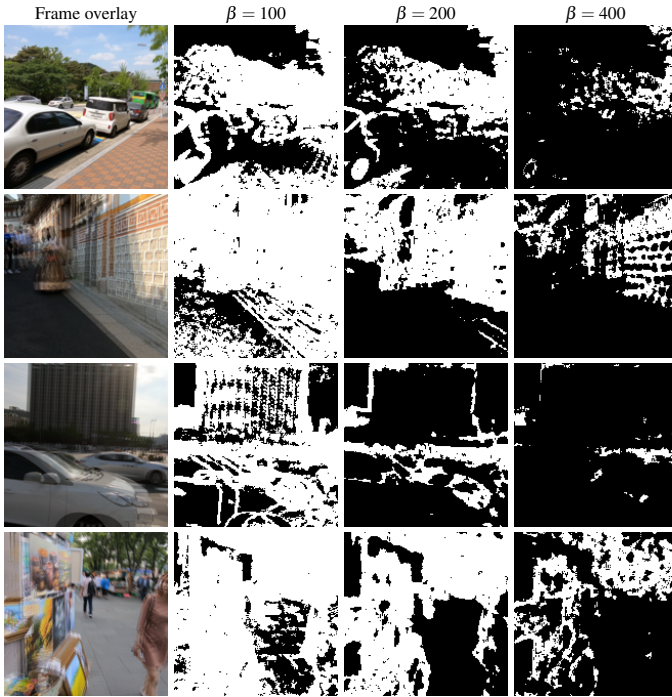


Figure 2: **Illustrations of selected query pixels.** The proposed GLSA module learns where to focus alignment, depending on the weight of the sparsity constraint β . Higher β s force the restriction of the alignment only where strictly needed, saving computation. Stationary and smooth regions (e.g. sky, ground) are likely to be skipped.

the alignment module operates in feature space - more precisely, after the frame encoder, see Sec. 3 in the main paper - we hereby employ it to warp the input frames to ease the representation. As such, we compute the attention matrix as in Eq. 3 in the paper. We then replace the values $\mathbf{v}_{sup} \in \mathbb{R}^{hw \times d_v}$ with a flattened version of the support frame, $\mathbf{x}_{sup} \in \mathbb{R}^{hw \times c}$. This procedure allows to warp the frame itself.

Fig. 4 illustrates some exemplars of alignment on the REDS4 dataset, for which we have *i.* a reference frame, *ii.* a misaligned support frame and *iii.* the same support frame after the alignment operation defined above. The figure represents the frame overlay and the norm of the residual between the reference and the support frames, both before and after alignment. It can be noticed how the overlay between reference and support frames before the alignment contains significant blurred regions due to motion. Contrarily, the overlay after alignment contains much less blurred regions, testifying the support frame has been warped to overlap to the reference frame. Similar considerations can be drawn by analyzing the residuals: despite the fact that some edge regions after alignment still exhibit some, GLSA is overall successful in reducing the spatial misalignment between pixels of the support and reference frames.

References

- [1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [2] Ping Hu, Fabian Caba, Oliver Wang, Zhe Lin, Stan Sclaroff, and Federico Perazzi. Temporally distributed networks for fast video semantic segmentation. In *CVPR*, 2020.
- [3] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *CVPRW*, 2019.

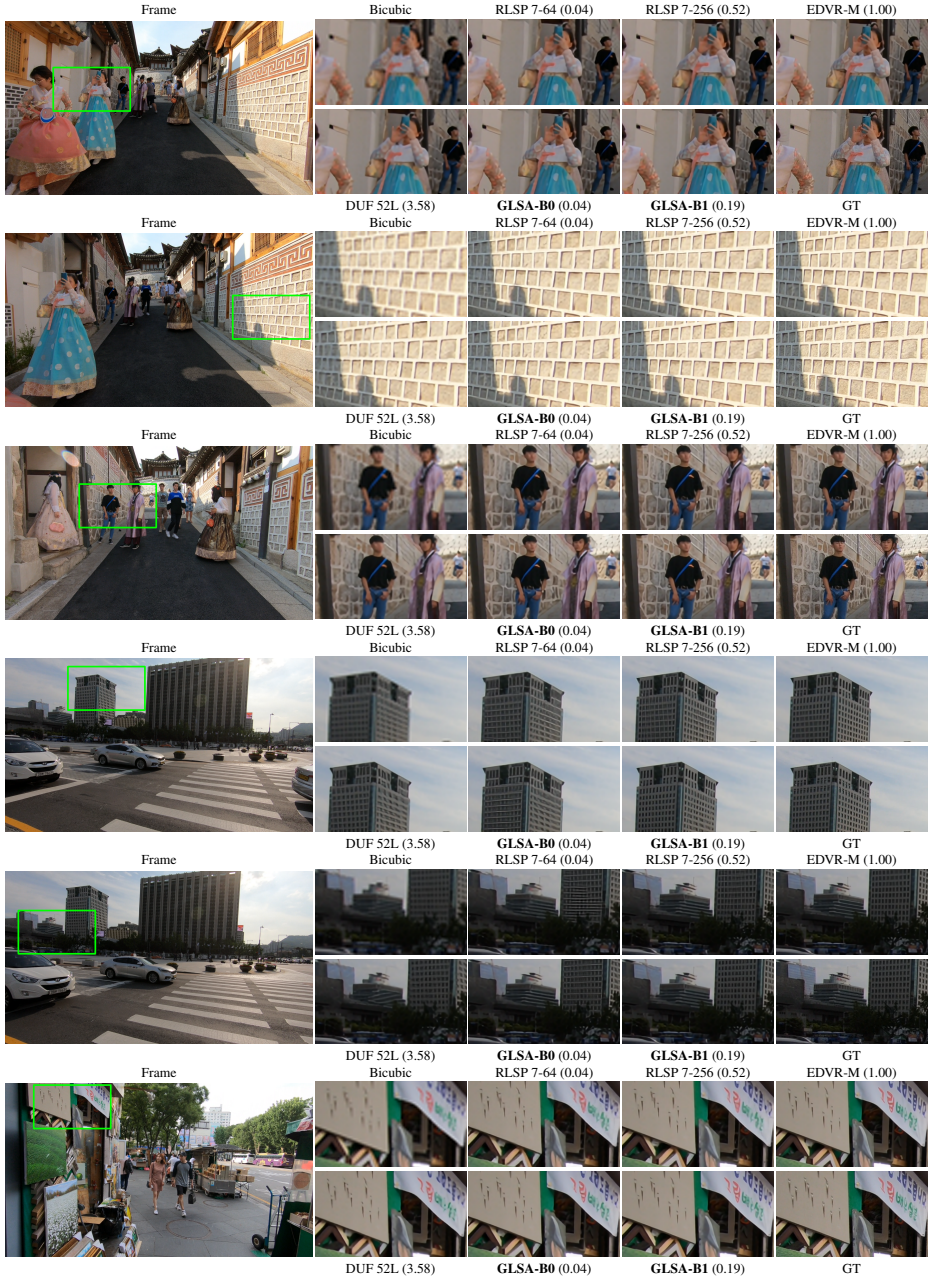


Figure 3: **Qualitative comparison on REDS4.** For comparisons in efficiency we report for each model, between brackets, the fraction of the GMACs it requires, compared to EDVR-M. We also refer the reader to Tab. 1 in the main paper.

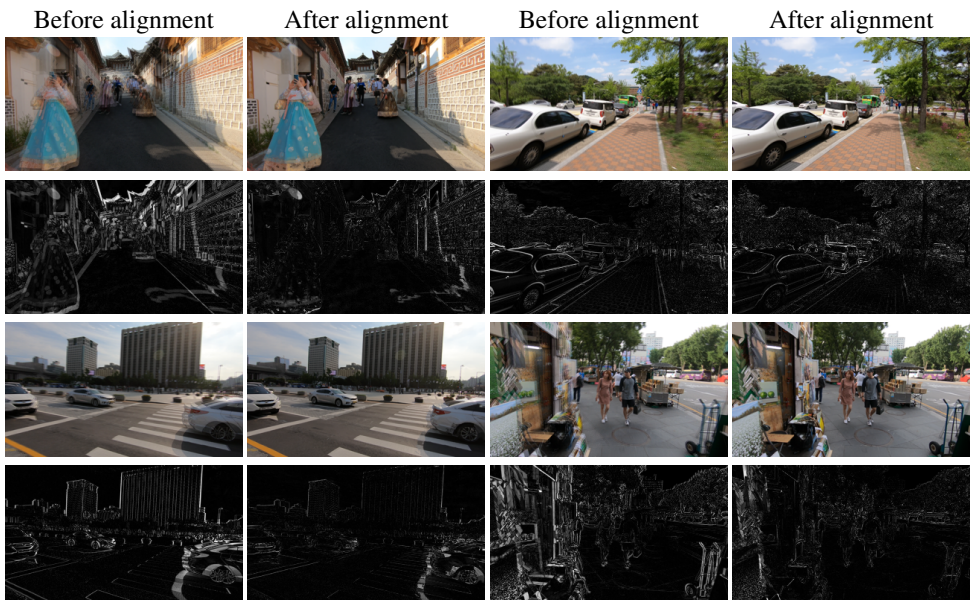


Figure 4: **Examples of GLSA alignment.** For 4 clips of REDS4, we represent frame overlay and residual between reference and support frames, both before and after the latter has been aligned via GLSA. The reduction in blur in frame overlays and in residuals advocates for the alignment capabilities of our proposed model. See Sec. 6 for details on how these visualizations were created.