Arbitrary-Scale Video Super-Resolution with Structural and Textural Priors

Wei Shang[1,2], Dongwei Ren[1⋆], Wanying Zhang[1],
Yuming Fang[3], Wangmeng Zuo[1], and Kede Ma[2]

[1] School of Computer Science and Technology, Harbin Institute of Technology
[2] Department of Computer Science, City University of Hong Kong
[3] Jiangxi University of Finance and Economics

Background

Background

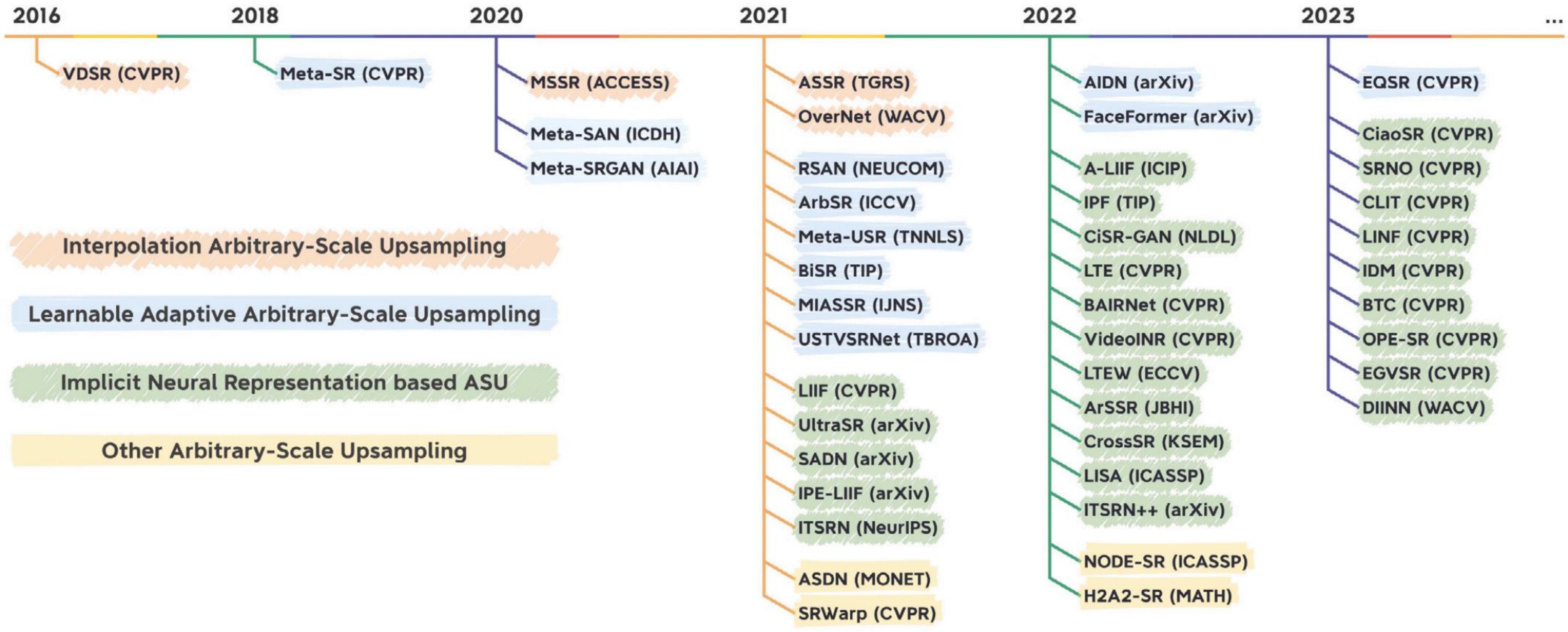


Fig. 4. Timeline of the development of deep learning-based arbitrary-scale super-resolution methods.

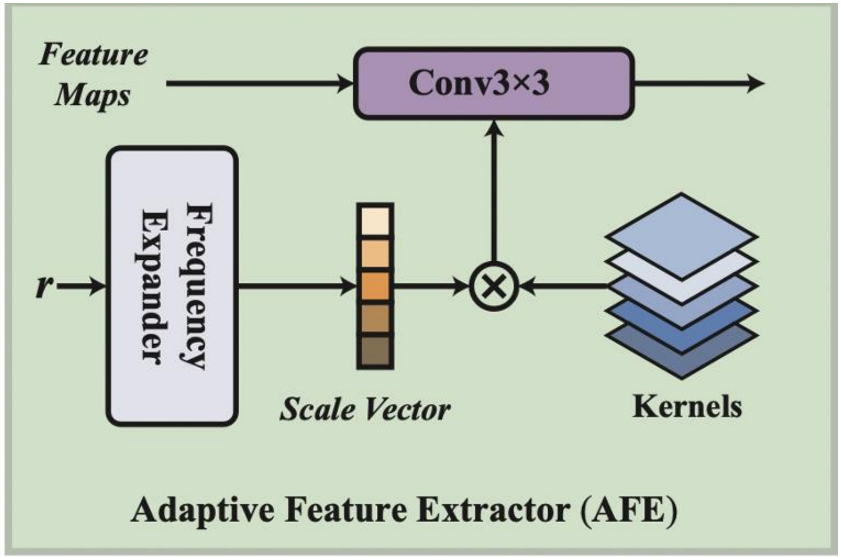
Background

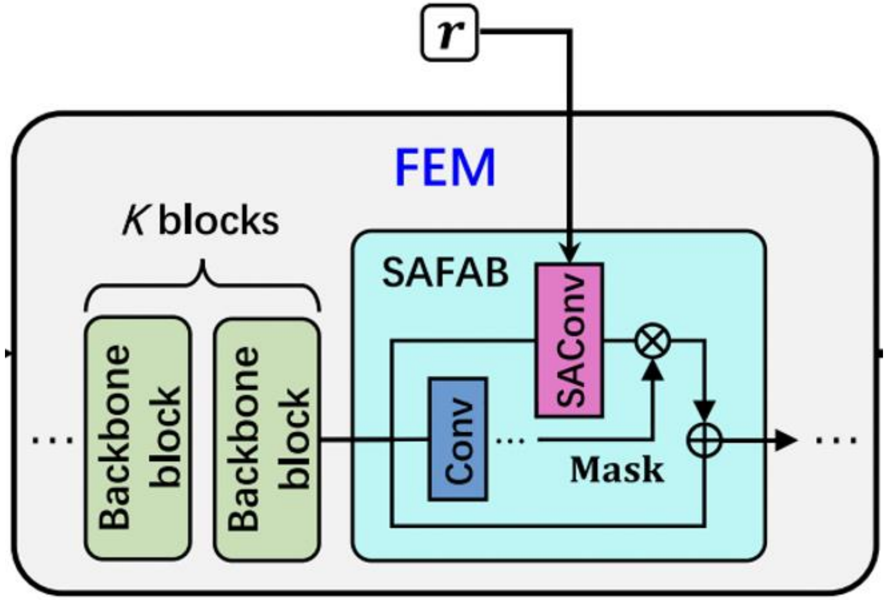
Problems:
1. Unable to leverage the temporal information between video frames

2. Existing ASVSR methods take only two frames as input, and cannot model long-range dependencies effectively, which also leads to worse temporal consistency.

3. Implicit neural representation will be impractical for resource-limited applications

4. Scale-aware convolutions, where the filters are dynamically customized based on scale factors.





Solution

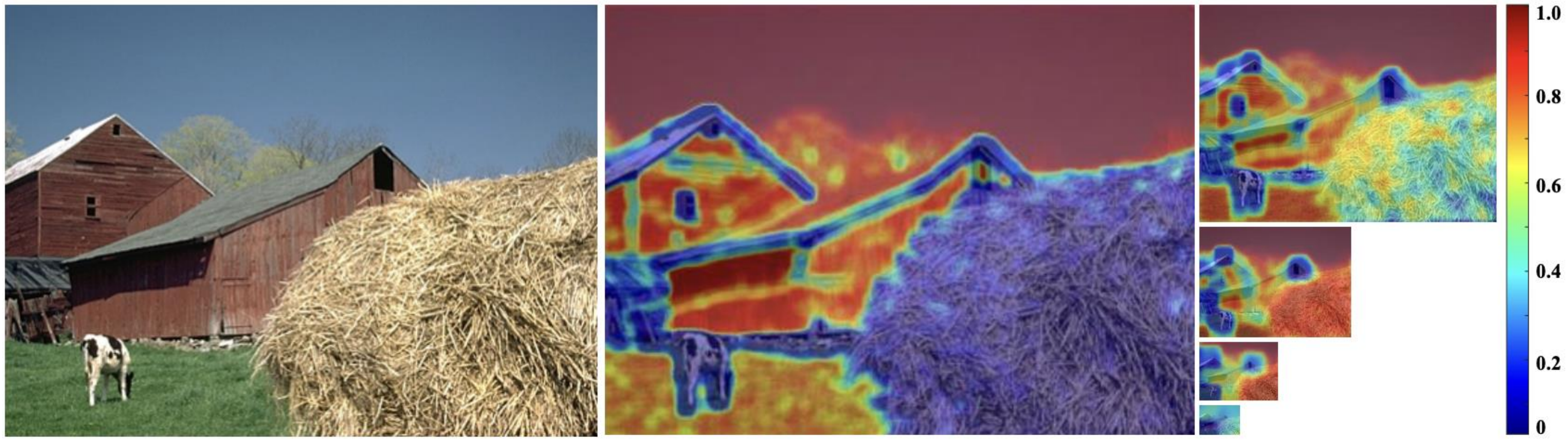


Fig. 1: Visualization of our multi-scale structural and textural prior derived from the pre-trained VGG network. A warmer color indicates a higher probability that the local patch at a given scale will be perceived as visual texture. Image borrowed from [12] with permission.

In summary, our main technical contributions include

- A strong baseline, B-AVSR, that is a nontrivial combination of three variants of elementary building blocks in literature [6,53,59],
- A high-performing AVSR algorithm, ST-AVSR, that leverages a powerful multi-scale structural and textural prior, and
- A comprehensive experimental demonstration, that ST-AVSR significantly surpasses competing methods in terms of SR quality on different test sets, generalization ability to unseen scales and degradation models, as well as inference speed.

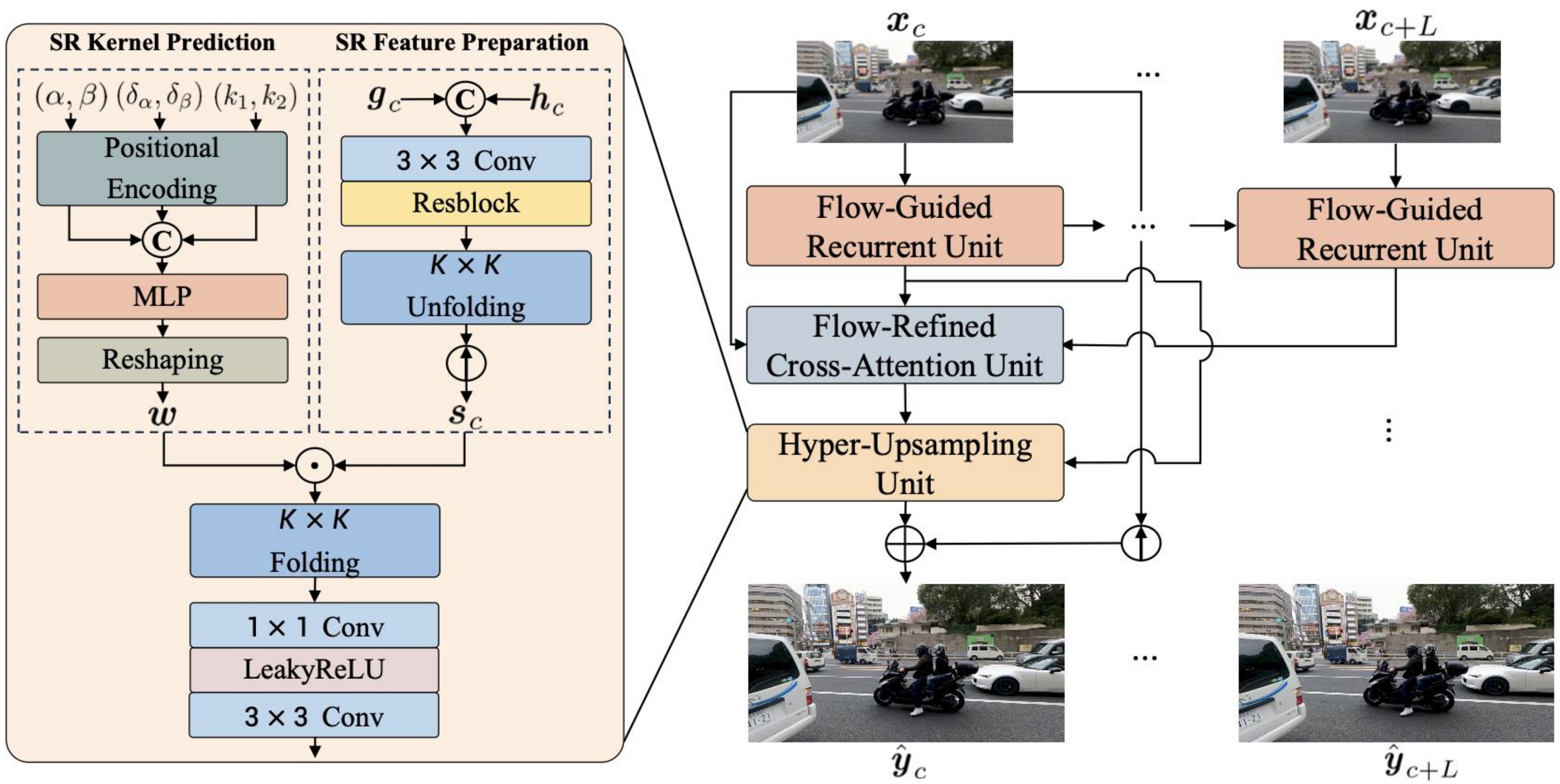


Fig. 2: System diagram of B-AVSR, which reconstructs an arbitrary-scale HR video $\hat{\boldsymbol{y}}$ from an LR video input $\boldsymbol{x}$. B-AVSR is composed of three variants of elementary building blocks: 1) a flow-guided recurrent unit to aggregate features from previous frames, 2) a flow-refined cross-attention unit to select features from future frames (see also Fig. 3), and 3) a hyper-upsampling unit to prepare SR features and predict SR kernels for HR frame reconstruction. ST-AVSR is built on top of B-AVSR by replacing all instances of $\boldsymbol{x}$ with the multi-scale structural and textural prior $\boldsymbol{p}$ (see the detailed text description in Sec. 3.4).

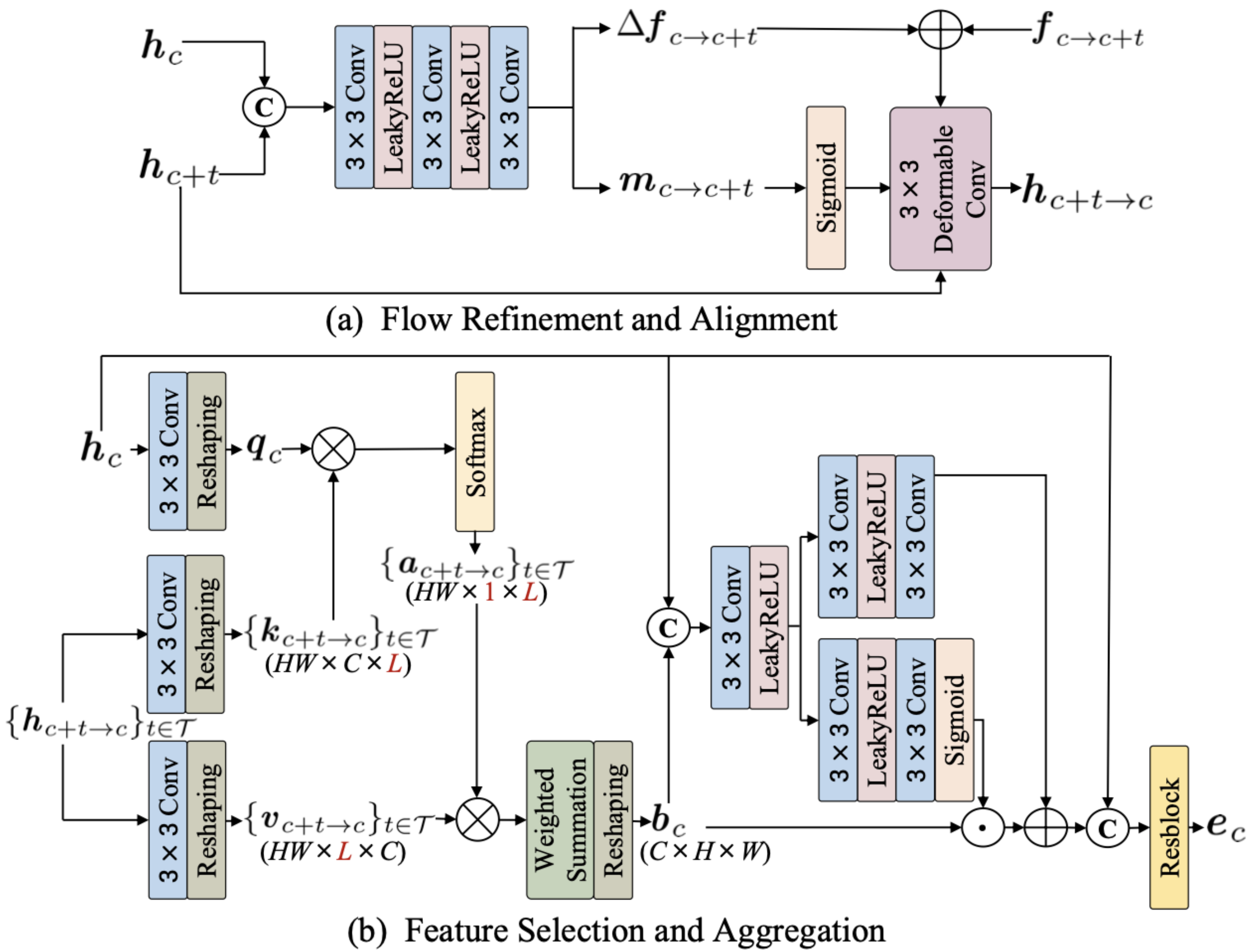


Fig. 3: Computational structure of the flow-refined cross-attention unit.

Training Pipline

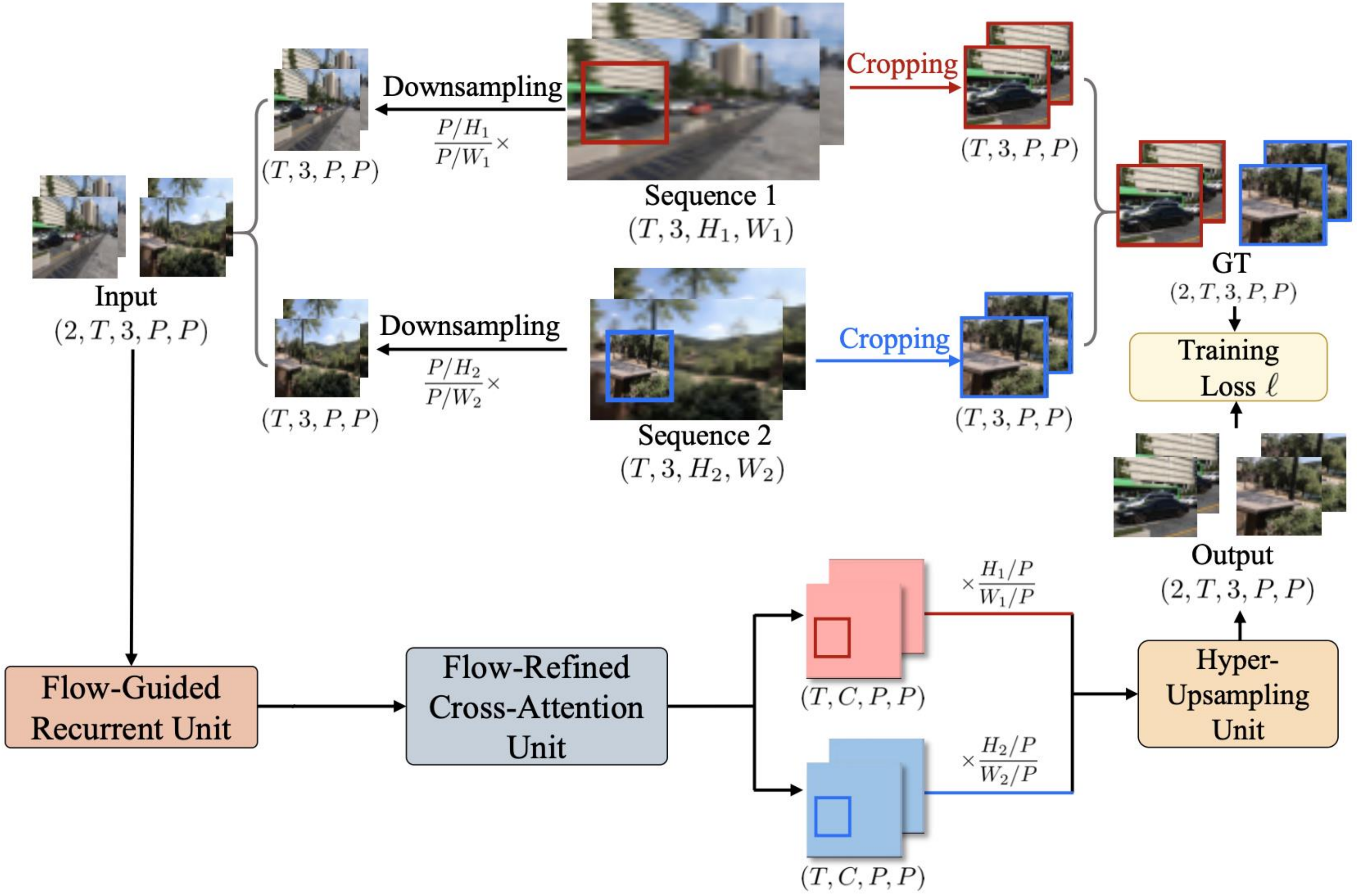


Fig. 1: Data pre-processing and training pipeline for B-AVSR and ST-AVSR.

Quantitative comparison

Table 1: Quantitative comparison with state-of-the-art methods for **arbitrary-scale video SR** on the REDS validation set (PSNR ↑ / SSIM ↑ / LPIPS ↓). Bold indicates the best performance.

Methods: Backbones	Methods: Upsampling Modules	Scale: ×2	×3	×4	×6	×8
Bicubic		31.51/0.911/0.165	26.82/0.788/0.377	24.92/0.713/0.484	22.89/0.622/0.631	21.69/0.574/0.699
ArbSR [35]		34.48/0.942/0.096	30.51/0.862/0.200	28.38/0.799/0.295	26.32/0.710/0.428	25.08/0.641/0.492
EQSR [36]		34.71/0.943/0.082	30.71/0.867/0.194	28.75/0.804/0.283	26.53/0.718/0.391	25.23/0.645/0.459
RDN [41]	LTE [17]	34.63/0.942/0.093	30.64/0.865/0.204	28.65/0.801/0.289	26.46/0.714/0.410	25.15/0.660/0.488
	CLIT [5]	34.63/0.942/0.092	30.63/0.865/0.204	28.63/0.801/0.290	26.43/0.714/0.400	25.14/0.661/0.467
	OPE [33]	34.05/0.939/0.082	30.52/0.864/0.199	28.63/0.800/0.293	26.37/0.711/0.421	25.04/0.655/0.504
SwinIR [18]	LTE [17]	34.73/0.943/0.091	30.73/0.866/0.200	28.75/0.804/0.284	26.56/0.718/0.403	25.24/0.669/0.480
	CLIT [5]	34.63/0.942/0.093	30.64/0.865/0.205	28.64/0.802/0.291	26.45/0.715/0.400	25.15/0.662/0.466
	OPE [33]	33.39/0.935/0.081	29.40/0.820/0.217	28.49/0.785/0.292	26.30/0.698/0.398	25.01/0.648/0.487
VideoINR [9]		31.59/0.900/0.144	30.04/0.852/0.197	28.13/0.791/0.263	25.27/0.687/0.374	23.46/0.619/0.470
MoTIF [7]		31.03/0.898/0.100	30.44/0.862/0.186	28.77/0.807/0.260	25.63/0.698/0.369	25.12/0.664/0.467
Ours		**36.91/0.969/0.041**	**33.41/0.937/0.066**	**31.03/0.897/0.114**	**27.89/0.812/0.222**	**26.04/0.746/0.298**

Table r1. Comparison of complexity and inference time.

Methods	ArbSR	EQSR	RDN			SwinIR			VideoINR	MoTIF	Ours
			LTE	CLIT	OPE	LTE	CLIT	OPE			
Complexity (GFLOPs)	887.3	1743.2	2011.3	7341.9	1003.7	1692.8	7022.3	684.0	1676.5	2826.2	**296.8**
Inference time (s)	0.6510	0.9211	0.5194	1.6549	0.2656	0.7289	1.9275	0.4379	0.6764	1.1320	**0.1010**

Table 2: Quantitative comparison with state-of-the-art methods for **arbitrary-scale video SR** on the Vid4 dataset (PSNR ↑ / SSIM ↑ / LPIPS ↓). The inference time is averaged over the Vid4 dataset for ×4 SR.

Methods: Backbones	Methods: Upsampling Modules	Scale: $\times\frac{2.5}{3.5}$	Scale: $\times\frac{4}{4}$	Scale: $\times\frac{7.2}{6}$	Inference time (s)
Bicubic		23.00/0.728/0.396	20.96/0.617/0.498	18.73/0.463/0.691	—
ArbSR [35]		25.86/0.815/0.224	24.01/0.721/0.313	21.23/0.540/0.478	0.2955
EQSR [36]		26.24/0.826/0.210	24.16/0.730/0.300	**21.72**/0.573/0.443	0.4181
RDN [41]	LTE [17]	25.98/0.818/0.226	24.03/0.722/0.312	21.64/0.565/0.455	0.2363
RDN [41]	CLIT [5]	25.83/0.815/0.223	23.94/0.721/0.312	21.62/0.563/0.458	0.7805
RDN [41]	OPE [33]	25.77/0.818/0.217	23.98/0.719/0.317	21.60/0.559/0.483	0.1242
SwinIR [18]	LTE [17]	26.43/0.826/0.217	24.09/0.727/0.305	**21.72**/0.570/0.448	0.3332
SwinIR [18]	CLIT [5]	25.89/0.818/0.224	24.00/0.724/0.314	21.65/0.565/0.457	0.9016
SwinIR [18]	OPE [33]	25.55/0.801/0.221	23.93/0.711/0.320	21.58/0.551/0.471	0.2008
VideoINR [9]		23.02/0.715/0.203	24.34/0.741/0.249	20.80/0.536/0.431	0.2364
MoTIF [7]		23.55/0.734/0.209	24.52/0.746/0.261	20.94/0.546/0.426	0.4053
Ours		**29.09**/**0.913**/**0.069**	**26.16**/**0.852**/**0.127**	21.60/**0.668**/**0.306**	**0.0495**

Visual comparison

Fig. 4: Visual comparison for arbitrary-scale SR models on the REDS dataset. Please zoom in for better view.

Visual comparison

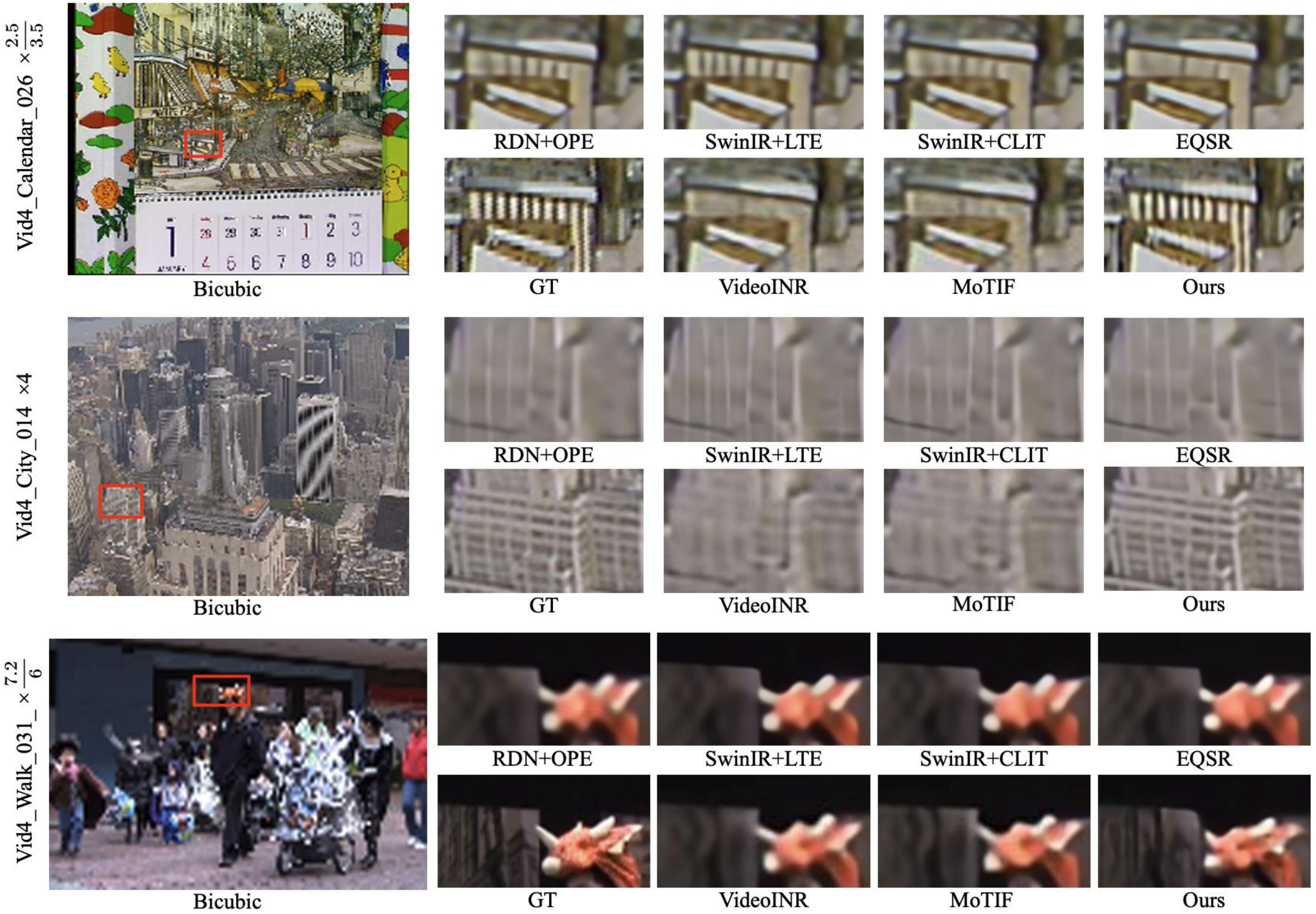


Fig. 5: Visual comparison for arbitrary-scale SR models on the Vid4 dataset.

x2.5_3.5_walk

x4_calendar

x8_000

Code :

More Video Results :

Thank you !