Spiking Meets Attention: Efficient Remote Sensing Image Super-Resolution with Attention Spiking Neural Networks

Yi Xiao[1] **Qiangqiang Yuan**[2*] **Kui Jiang**[3] **Wenke Huang**[2] **Qiang Zhang**[4]
Tingting Zheng[3] **Chia-Wen Lin**[5] **Liangpei Zhang**[2]

[1]School of Computer and Artificial Intelligence, Zhengzhou University
[2]Wuhan University [3]Harbin Institution of Technology [4]Dalian Maritime University
[5]National Tsinghua University
yixiao@zzu.edu.cn

Abstract

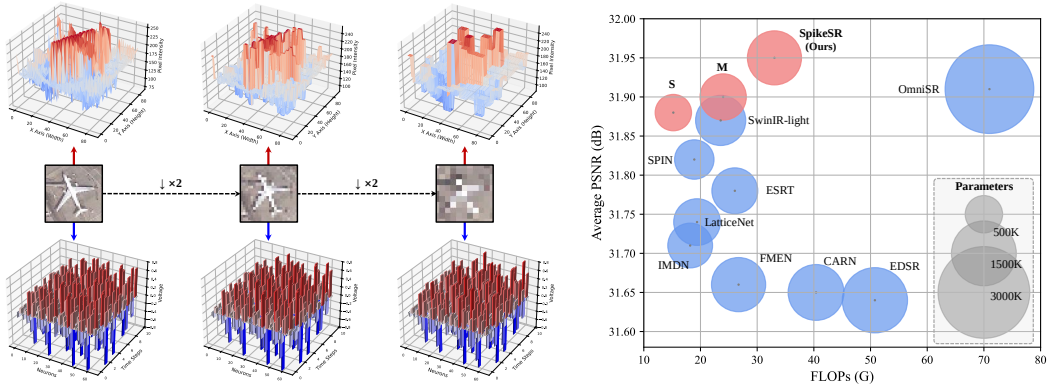
Spiking neural networks (SNNs) are emerging as a promising alternative to traditional artificial neural networks (ANNs), offering biological plausibility and energy efficiency. Despite these merits, SNNs are frequently hampered by limited capacity and insufficient representation power, yet remain underexplored in remote sensing image (RSI) super-resolution (SR) tasks. In this paper, we first observe that spiking signals exhibit drastic intensity variations across diverse textures, highlighting an active learning state of the neurons. This observation motivates us to apply SNNs for efficient SR of RSIs. Inspired by the success of attention mechanisms in representing salient information, we devise the spiking attention block (SAB), a concise yet effective component that optimizes membrane potentials through inferred attention weights, which, in turn, regulates spiking activity for superior feature representation. Our key contributions include: 1) we bridge the independent modulation between temporal and channel dimensions, facilitating joint feature correlation learning, and 2) we access the global self-similar patterns in large-scale remote sensing scenarios to infer spatial attention weights, incorporating effective priors for realistic and faithful reconstruction. Building upon SAB, we proposed SpikeSR, which achieves state-of-the-art performance across various remote sensing benchmarks such as AID, DOTA, and DIOR, while maintaining high computational efficiency. Code of SpikeSR will be available at https://github.com/XY-boy/SpikeSR.

High-resolution remote sensing images (RSIs) contain fine-grained object structures and textures, which are critical for accurate interpretation in downstream tasks [42, 11, 6]. However, limited by the intrinsic resolution of airborne sensors, RSI can merely capture partial spatial details, resulting in suboptimal scene representation and visual quality. Image super-resolution (SR) aims to alleviate this problem by reconstructing high-resolution (HR) images from low-resolution (LR) observations [53, 60]. Despite this, SR remains a challenging ill-posed issue, as a degraded input may correspond to multiple plausible outputs.

Early efforts rely on hand-crafted priors to tame the ill-posedness, *e.g.*, nonlocal mean [67, 7] and gradient profile [48], but they are often trapped in limited performance and scalability. Recent advances in artificial neural networks (ANNs), *e.g.*, CNNs and Transformers, have witnessed remarkable progress in SR with large-capacity models [49, 50, 3, 4, 71]. However, they often come with a

*Corresponding Author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).



(a) Pixel Intensity (top) and Neuron Voltage (down) Maps **(b) Performance and Efficiency Comparison**

Figure 1: (a) The visualization of pixel intensity and neuron voltage in images under various degradation factors reveals important insights. The pixel intensity map illustrates that the high-frequency components of the image tend to be smooth, indicating a reduction in sharp details during progressive downsampling. Neuron intensity maps, derived from a LIF model [37, 14], show that high-frequency details persist with drastic fluctuations, suggesting that the neurons remain in an active state. (b) FLOPs and PSNR performance comparison. The circle sizes represent the number of parameters. Our SpikeSR outperforms SOTA efficient SR methods with high efficiency. PSNR results are averaged on the AID, DOTA, and DIOR datasets.

trade-off of increased computational overhead and growing storage costs, making them less efficient in practical scenarios, particularly when reconstructing large-scale RSIs.

More recently, brain-inspired spiking neural networks (SNNs), as the third generation of neural networks, have emerged as a promising alternative for energy-efficient intelligence [25, 59, 40]. Different from ANNs that encode features as continuous values, SNNs can emulate biological communication with discrete spiking signals and propagate them by neurons, thus enjoying lower power consumption. As depicted in Fig. 1(a), our experiments reveal a novel finding that spiking neurons maintain an active learning state across LR RSIs, even in severely damaged textures. Specifically, we observed that degraded RSIs exhibit smoothed pixel intensities and obscured sharp details, posing a significant challenge to characterize high-frequency representations. In contrast, spiking signals retain drastic responses and pronounced spike rates, highlighting that neurons remain in an active learning state. This naturally arises a question: *Can SNNs leverage their inherent properties to handle image degradation for efficient yet high-quality RSI SR?*

In fact, to effectively grasp complex and diverse spatial details in RSIs, the network must possess adequate capacity and representation power. Unfortunately, there are two critical challenges when adapting SNNs for SR tasks. Firstly, **spiking activity in SNNs inevitably causes pixel-wise information loss,** which hampers the representation capacity of SNNs, especially when the network deepens. This stems from the discrete nature of binary spiking signals, leading to undesirable spiking degradation problems [61, 15]. Secondly, **SNNs remain constrained by suboptimal membrane potential dynamics,** restricting effective exploration of global context during spiking communications. This necessitates a customized strategy to optimize membrane potentials, but is barely explored before.

To address these limitations, we propose SpikeSR, an SNN-based framework inspired by human visual attention mechanisms, which can actively represent image degradation and modulate synaptic weights to focus on salient regions, which, in turn, regulate the spiking activity for improved capacity and representation power. Specifically, SpikeSR employs a concise yet effective spiking attention block (SAB) to optimize feature emphasis through spiking response dynamics, which integrates three key innovations: 1) the combination of CNN and SNN layers to mitigate information loss induced by discrete spiking activity; 2) introducing hybrid dimension attention (HDA) to recalibrate spiking response across both temporal and channel dimensions, facilitating a joint feature correlations learning; 3) accessing global self-similarity patterns in RSIs to infer spatial attention weights, incorporating effective priors for realistic and faithful reconstruction. Compared to state-of-the-art (SOTA) ANN-based efficient SR models, our SpikeSR demonstrates lower model complexity and superior performance, as shown in Fig. 1(b).

Our contributions are summarized as follows:

- We pioneer an attention spiking neural network for efficient SR of RSIs, providing a new perspective on developing efficient models in large-scale Earth observation scenarios.
- We devise a concise yet effective SAB, which mitigates the information loss and regulates membrane potentials of spiking activity for improved representation of SNNs.
- Extensive experimental results on various remote sensing datasets demonstrate that our SpikeSR achieves competitive SR performance against SOTA ANN-based methods.

1 Related Work

Deep Networks for SR. Inspired by the pioneering SRCNN [10], CNN-based SR methods have achieved remarkable progress, dominating the field for years. They mainly elaborated on the network design to tame the ill-posedness, with notable advances in residual connections [21, 32] and attention mechanisms [69, 39]. However, these methods often suffer from high computational complexity, *e.g.*, exhaustive non-local modeling [28, 38], making them less efficient in large-scale RSIs.

Recently, transformer-based SR models have demonstrated impressive performance, benefiting from their ability to model long-range dependencies. IPT [2] first introduces Transformers in SR field, but requires massive parameters and laborious pre-training processes. SwinIR [31] effectively reduces the model size by partitioning the image into smaller windows when applying multi-head attention mechanisms, while maintaining favorable performance. Against transformer, Mamba-based SR methods achieve comparable global model capacity with linear complexity [18, 57, 17]. Despite these advancements, advanced SR models are often trapped by rising computational overhead and growing storage costs, posing significant concerns in real-world applications, particularly in remote sensing scenarios.

Efficient SR Models. To reduce computational budget, CARN [1] utilizes grouped convolutions and a cascading mechanism to improve the residual architecture. IMDN [20] progressively distills useful information during feature extraction and applies network pruning to further decrease complexity. FMEN [12] optimizes residual modules to accelerate inference. In Transformer-based SR methods, SPIN [66] enhances long-range modeling by combining self-attention with pixel clustering, facilitating interactions between superpixels. HiT-SR [68] expands the self-attention receptive field by applying different window sizes of hierarchal layers. Despite these successes, there is still room to further boost SR performance. Moreover, the potential of energy-efficient SNNs for SR tasks remains largely unexplored.

Spiking Neural Networks. Recent advances in neuromorphic computing have shown the great potential of SNNs in computational efficiency and power as CNNs [63, 64]. Currently, SNNs have been successfully applied to various tasks, such as image classification [25, 44], object detection and tracking [22, 59], optical flow estimation [26, 40], *etc*.

A common solution to build SNNs is converting pre-trained ANN models [8, 62]. Li *et al.* [30] proposed a layer-wise calibration to minimize activation mismatch during conversion. Ding *et al.* [9] replaced ReLU with the rate norm layer, enabling direct conversion from a trained ANN to an SNN. Stockl *et al.* [46] used time-varying multi-bit spikes to better approximate activation functions. However, conversion-based methods face accuracy gaps and high latency due to extensive time-step simulations, resulting in increased latency and energy consumption.

An alternative involves using agent gradient functions for continuous relaxation of non-smooth spike activities, enabling direct training via backpropagation through time. Lee *et al.* [27] treated membrane potential as a signal to overcome discontinuities, enabling direct training from spikes. Wang *et al.* [54] introduced an iterative LIF model and proposed spatiotemporal backpropagation based on approximate peak activity derivatives. Later, Zheng *et al.* [70] proposed temporal delay batch normalization, which significantly enhanced the depth of SNNs. To bridge the performance gap between ANNs and SNNs, some methods borrowed insights from CNNs, applying residual learning [15, 19] and attention mechanisms [65, 41] to SNNs. Nonetheless, there has been limited exploration of pixel-level regression tasks, such as SR.

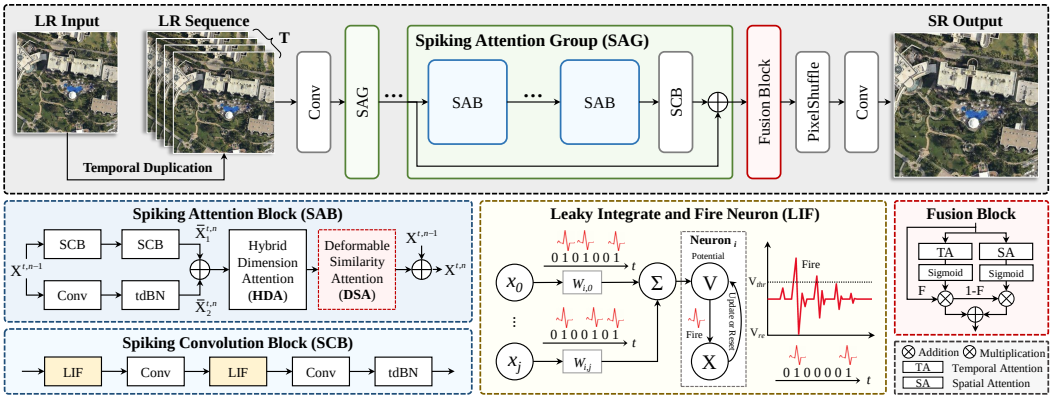


Figure 2: Overall network architecture of SpikeSR. The LR input is replicated along the temporal dimension and then processed through a convolution to extract shallow features. The core module of SpikeSR is SAG, which employs SABs to capture deep spiking representations. Each SAB contains three main components: (1) SCB, (2) HDA, and (3) DSA. The fusion block (FB) aggregates the spatial-temporal sequences, and pixelshuffle is used to reconstruct the SR output.

2 Method

The architecture of the proposed SpikeSR is illustrated in Fig. 2, which mainly consists of SAGs. Before SAGs, we utilize a 3×3 convolution to extract high-dimensional features from the LR input. These features are then processed through m stacked SAGs to explore deeper representations. Each SAG includes n SABs, a SCB, and a residual connection. In the SCB, leaky integrate-and-fire (LIF) neurons [37, 14] are used to convert the inputs into binary spike sequences (*i.e.*, 0 or 1). As shown in Fig. 2, the output of the LIF neuron is 1 when the membrane potential exceeds the threshold, and 0 otherwise. To optimize the membrane potential, we introduce HDA, which refines the spiking activity using an efficient temporal-channel joint attention [72]. Furthermore, the proposed DSA is employed to introduce global context for accurate SR. After the terminal SAG, an FB is utilized to convert the spike sequence features into continuous values. Finally, SpikeSR generates super-resolved output from the fused features by applying pixel-shuffle [43] and a 3×3 convolutional layer.

2.1 Spiking Attention Block

As evidenced in Fig. 1, regions degraded by different factors exhibit noticeable fluctuations when encoded by LIF, highlighting pronounced firing spike rates of neurons. This provides robust and latent informative spiking cues from LR images. Unlike ANNs that encode images into continuous decimal values, SNNs use discrete binary spike values for neuronal communication, and thus demonstrate undesirable information loss [24, 65], resulting in limited capacity to represent degraded LR images. To address this, the design philosophy of the SAB is focused on leveraging CNNs and attention mechanisms to regulate membrane potentials, facilitating high-quality feature representation for SR, which in turn affects the spiking activity.

As shown in Fig. 2, in particular, the output of the n-th SAB at the t-th time step is denoted as $\mathbf{X}^{t,n}$, and can be obtained by the following:

$$\mathbf{X}^{t,n} = \mathbf{X}^{t,n-1} + \mathrm{DSA}(\mathrm{HDA}(\bar{\mathbf{X}}_1^{t,n} + \bar{\mathbf{X}}_2^{t,n})), \tag{1}$$

where $\bar{\mathbf{X}}_1^{t,n}$ and $\bar{\mathbf{X}}_2^{t,n}$ are two feature representations obtained from parallel branches, defined by:

$$\begin{aligned} \bar{\mathbf{X}}_1^{t,n} &= \mathrm{SCB}(\mathrm{SCB}(\mathbf{X}^{t,n-1})), \\ \bar{\mathbf{X}}_2^{t,n} &= \mathrm{tdBN}(\mathrm{Conv}(\mathbf{X}^{t,n-1})), \end{aligned} \tag{2}$$

where Conv represents a 3×3 convolution layer, and tdBN means the threshold-dependent batch normalization.

Different from previous works that focus solely on separate temporal and channel modulation [61, 65, 45], SAB adheres to temporal-channel joint attention [72] to realize joint adjustment of the spike response in HDA, effectively achieving interdependencies between the temporal and channel scopes. More details of HDA can be found in the Appendix.

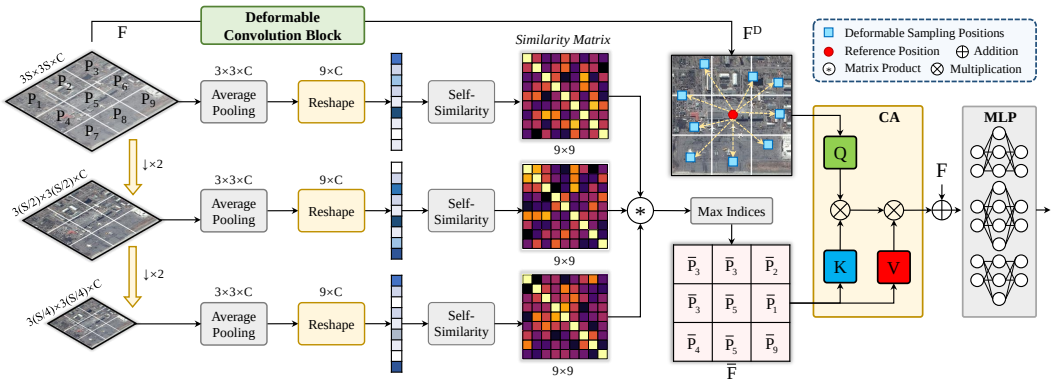


Figure 3: The illustration of our DSA. Note that we set the diagonal elements of the similarity matrix to zero before selecting the indices of the highest scores. The deformable convolution operates at the patch level, alleviating the mismatch between the most similar patches.

2.2 Deformable Similarity Attention

Non-local self-similarity has been recognized as an effective prior for SR tasks [47]. However, existing non-local attention mechanisms are computationally expensive due to exhaustive non-local operations, which impedes their efficiency in large-scale RSIs. In contrast, the proposed DSA efficiently grasps complex self-similar patterns in RSIs at the patch level to infer intricate spatial weights. Then, we utilize the cross-attention (CA) paradigm to enhance long-range communication, facilitating the fusion of useful context.

The details of DSA are shown in Fig. 3. Considering that the object scale exhibits explicit diversity in RSIs, the input feature F is downsampled using bilinear interpolation, forming a multi-scale feature pyramid. For clarity, we demonstrate this process by dividing the initial features into 9 patches. Following the design in [33], the final DSA exploits a cascaded patch division strategy. Specifically, each patch is first average-pooled to capture its spatial characteristics, then reshaped and subjected to self-similarity computation, yielding a similarity matrix. The final self-similarity scores are fused via matrix multiplication to enhance the multi-scale representation. The best-matching patch $\bar{\mathbf{P}}_i$ with $\mathbf{P}_i$ can be obtained by:

$$\bar{\mathbf{P}}_i = \underset{\mathbf{P}_j}{\operatorname{argmax}}\, E(\mathbf{P}_i)^{\mathrm{T}} E(\mathbf{P}_j), \quad j \neq i, \tag{3}$$

where $\bar{\mathbf{P}}_i$ is the patch in $\bar{\mathbf{F}}$, and E means the operation of average pooling and feature reshaping. We adopt the Gumbel-Softmax [52] to achieve the non-differentiable argmax function.

Although matched patches contain highly relevant similarity, they are inevitably subject to mismatches and geometric transformations. Hence, we use deformable convolution (DConv) to reduce the generation of hallucinated textures. The deformable feature $\mathbf{F}^{\mathrm{D}}$ at location p_0 is computed as follows:

$$\mathbf{F}^{\mathrm{D}}(p_0) = \sum_{p_m \in \mathcal{R}} \omega(p_m) \cdot \mathbf{F}(p_0 + p_m + \Delta p_m), \tag{4}$$

where $\omega(p_m)$ is the convolution weight at relative location p_m, Δp_m is a 2D vector that represents the learnable offsets, $\mathcal{R}$ is a regular grid that determines the receptive field of the convolution kernel. For a 3×3 kernel, $\mathcal{R} = \{(-1,-1),(-1,0),\cdots,(1,1)\}$. To fuse the self-similar features $\mathbf{F}^{\mathrm{D}}$ with $\bar{\mathbf{F}}$, we embed $\mathbf{F}^{\mathrm{D}}$ to $\mathbf{Q}$, and $\bar{\mathbf{F}}$ to $\mathbf{K}$, $\mathbf{V}$ using fully connected layers, then perform aggregation by:

$$\bar{\mathbf{V}} = \operatorname{softmax}(\mathbf{Q}\mathbf{K}^{\mathrm{T}}/\sqrt{d})\mathbf{V}, \tag{5}$$

Finally, the fused features are summarized with the original features $\mathbf{F}$ and fed into the multilayer perceptron (MLP) to obtain the final output:

$$\tilde{\mathbf{F}} = \mathrm{MLP}(\mathbf{F} + \bar{\mathbf{V}}). \tag{6}$$

Table 1: Quantitative comparison of SpikeSR with SOTA methods on three remote sensing datasets. FLOPs are measured corresponding to an LR image of 160×160 pixels. Note that we set $\mathrm{T} = 1$ to evaluate the model complexity of SpikeSR for fair comparison.

Methods	#Param.	FLOPs	AID [56]		DOTA [55]		DIOR [29]		Average	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bicubic	-	-	28.86	0.7382	31.16	0.7947	28.57	0.7432	29.53	0.7587
SRCNN [10]	20K	0.512G	29.70	0.7741	32.10	0.8264	29.49	0.7768	30.43	0.7924
VDSR [21]	667K	17.08G	30.44	0.8004	33.22	0.8569	30.36	0.8036	31.34	0.8203
EDSR [32]	1518K	50.77G	30.65	0.8086	33.64	0.8648	30.63	0.8116	31.64	0.8283
CARN [1]	1112K	40.39G	30.66	0.8068	33.66	0.8633	30.64	0.8102	31.65	0.8268
IMDN [20]	715K	18.18G	30.71	0.8076	33.70	0.8641	30.73	0.8115	31.71	0.8277
RFDN-L [34]	681K	16.49G	30.69	0.8074	33.73	0.8642	30.72	0.8114	31.71	0.8277
LatticeNet [36]	777K	19.39G	30.73	0.8089	33.75	0.8653	30.75	0.8126	31.74	0.8289
HNCT [13]	364K	8.48G	30.79	0.8104	33.83	0.8664	30.80	0.8136	31.81	0.8301
FMEN [12]	1046K	26.72G	30.65	0.8063	33.66	0.8631	30.66	0.8104	31.66	0.8266
RLFN [23]	544K	13.25G	30.70	0.8074	33.69	0.8636	30.70	0.8110	31.70	0.8273
ESRT [35]	752K	26.06G	30.77	0.8102	33.75	0.8668	30.81	0.8142	31.78	0.8304
SwinIR-light [31]	897K	23.56G	30.83	0.8114	33.94	0.8677	30.85	0.8149	31.87	0.8313
Omni-SR [51]	2803K	70.98G	30.89	**0.8142**	33.94	0.8695	30.89	0.8170	31.91	0.8336
NGswin [5]	995K	12.73G	30.79	0.8107	33.87	0.8667	30.79	0.8140	31.82	0.8305
SPIN [66]	555K	18.91G	30.78	0.8098	33.85	0.8673	30.82	0.8139	31.82	0.8303
HiT-SR [68]	792K	21.04G	30.87	0.8138	33.93	0.8689	30.89	0.8167	31.90	0.8331
SpikeSR-S (Ours)	472K	15.21G	30.86	0.8126	33.89	0.8687	30.89	0.8162	31.88	0.8325
SpikeSR-M (Ours)	763K	24.00G	30.88	0.8133	33.92	0.8689	30.90	0.8163	31.90	0.8328
SpikeSR (Ours)	1042K	33.05G	**30.91**	**0.8142**	**33.98**	**0.8700**	**30.95**	**0.8175**	**31.95**	**0.8339**

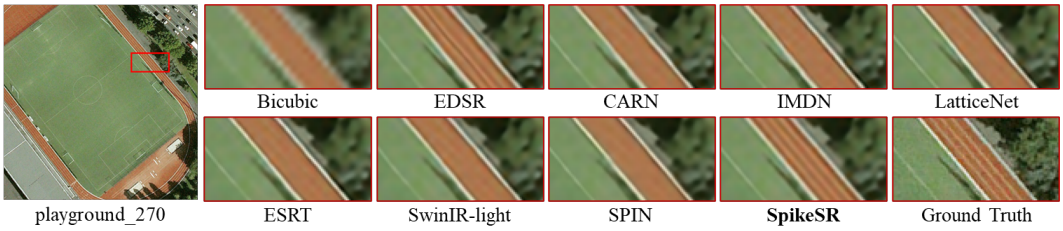


Figure 4: Qualitative comparison of SOTA efficient models for $\times 4$ SR task on AID test set.

2.3 Fusion Block

To transform discrete spiking sequences into continuous pixel values, a common approach is to apply mean sampling along the time dimension. However, this naive process may lead to the loss of crucial spatial details, potentially affecting the SR quality. Therefore, we introduce a fusion block that adaptively aggregates spiking sequences and mitigates information loss. Given an input spike input $\mathbf{Y}$, the computation process of FB can be formulated as:

$$\begin{aligned} \mathbf{Y}_1 &= \sigma(\mathrm{TA}(\mathbf{Y})) \otimes \mathbf{Y}, \\ \mathbf{Y}_2 &= \sigma(\mathrm{SA}(\mathbf{Y})) \otimes (1 - \mathbf{Y}_1), \end{aligned} \tag{7}$$

where TA and SA denote temporal and spatial attention [65], σ means a sigmoid function and $\otimes$ denotes feature multiplication. The final output of FB is obtained by summing $\mathbf{Y}_1$ and $\mathbf{Y}_2$.

3 Experiments

Datasets. We use the AID dataset [56] as the training set, a large-scale remote sensing benchmark for scene classification, consisting of 30 different scene categories. The AID dataset includes 10,000 HR images, where we randomly select 3,000 for training and 900 for validation. The LR samples are generated by bicubic downsampling. Following TTST [58], we also evaluate our method on the DOTA [55] and DIOR [29] datasets, which contain 900 and 1,000 images, respectively.

Implementation Details. During model training, the learning rate is fixed to 10^{-4}, and the training procedure stops after 1000 epochs with a batch size of 4. Adam optimizer is used with $\beta_1 = 0.9$

Table 2: Quantitative comparison of SpikeSR with SOTA methods on 30 scene types of AID datasets.

Scene types	EDSR [32]		CARN [1]		IMDN [20]		ESRT [35]		SwinIR-L [31]		SPIN [66]		HiT-SR [68]		SpikeSR	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Airport	29.93	0.8282	29.96	0.8264	30.00	0.8270	30.09	0.8292	30.14	0.8307	30.11	0.8295	30.21	0.8325	**30.27**	**0.8336**
Bare Land	36.94	0.8837	36.92	0.8829	36.93	0.8834	36.90	0.8840	36.99	0.8841	36.96	0.8843	**37.00**	**0.8846**	36.98	**0.8846**
Baseball Field	33.05	0.8765	33.06	0.8753	33.17	0.8763	33.21	0.8773	33.27	0.8782	33.14	0.8759	33.29	0.8791	**33.32**	**0.8794**
Beach	34.18	0.8727	34.27	0.8737	34.29	0.8739	34.35	0.8755	34.39	0.8756	34.36	0.8757	34.38	0.8762	**34.41**	**0.8764**
Bridge	32.93	0.8800	32.86	0.8774	32.93	0.8784	33.05	0.8803	33.15	0.8810	33.05	0.8803	33.14	0.8820	**33.27**	**0.8827**
Center	28.77	0.7921	28.71	0.7881	28.79	0.7892	28.88	0.7923	29.00	0.7954	28.88	0.7919	29.03	0.7974	**29.09**	**0.7985**
Church	26.30	0.7469	26.41	0.7449	26.46	0.7467	26.52	0.7489	26.59	0.7512	26.56	0.7507	26.64	0.7549	**26.70**	**0.7560**
Commercial	29.01	0.7940	29.11	0.7944	29.17	0.7958	29.22	0.7975	29.28	0.7996	29.27	0.7989	29.33	0.8021	**29.37**	**0.8033**
D-Residential	24.38	0.6839	24.56	0.6856	24.60	0.6864	24.67	0.6898	24.71	0.6924	24.63	0.6872	24.80	**0.6998**	**24.80**	0.6978
Desert	40.20	0.9268	40.22	0.9259	40.17	0.9264	40.06	0.9276	40.31	0.9275	40.24	0.9279	**40.27**	0.9282	40.25	**0.9283**
Farmland	35.00	0.8683	34.89	0.8656	34.94	0.8667	34.99	0.8679	35.02	0.8681	34.98	0.8678	35.09	0.8696	**35.09**	**0.8700**
Forest	29.85	0.7315	29.88	0.7304	29.90	0.7312	29.99	0.7365	29.98	0.7350	29.97	0.7350	30.05	**0.7395**	**30.05**	0.7380
Industrial	28.88	0.7931	28.84	0.7894	28.88	0.7904	28.98	0.7942	29.03	0.7959	28.98	0.7932	29.11	0.7998	**29.16**	**0.8007**
Meadow	34.63	0.7804	34.53	0.7769	34.55	0.7784	34.63	0.7814	34.66	0.7807	34.64	0.7813	34.58	0.7807	**34.68**	**0.7820**
M-Residential	28.34	0.7365	28.39	0.7347	28.42	0.7349	28.47	0.7370	28.56	0.7401	28.39	0.7334	**28.64**	**0.7443**	28.63	0.7436
Mountain	30.63	0.7885	30.70	0.7892	30.70	0.7895	30.74	0.7909	30.78	0.7921	30.75	0.7911	**30.79**	**0.7930**	**30.79**	**0.7930**
Park	30.54	0.8130	30.63	0.8136	30.65	0.8141	30.72	0.8169	30.76	0.8177	30.73	0.8162	30.81	0.8198	**30.82**	**0.8203**
Parking	27.25	0.8317	27.08	0.8245	27.23	0.8270	27.47	0.8352	27.42	0.8354	27.50	0.8363	27.70	**0.8435**	**27.72**	0.8424
Playground	35.37	0.8943	35.27	0.892	35.42	0.8929	35.47	0.8942	35.49	0.8946	35.45	0.8946	35.59	0.8968	**35.70**	**0.8976**
Pond	32.11	0.8542	32.10	0.8532	32.11	0.8534	32.17	0.8546	32.22	0.8553	32.15	0.8545	32.23	0.8561	**32.25**	**0.8563**
Port	28.50	0.8596	28.61	0.8593	28.67	0.8597	28.75	0.8620	28.81	0.8631	28.79	0.8624	28.85	0.8651	**28.94**	**0.8658**
Railway Station	28.72	0.7738	28.68	0.7699	28.77	0.7718	28.84	0.7744	28.92	0.7777	28.89	0.7759	28.97	0.7802	**29.02**	**0.7816**
Resort	28.52	0.7799	28.59	0.7791	28.62	0.7795	28.68	0.7819	28.74	0.7837	28.66	0.7801	28.78	0.7864	**28.82**	**0.7869**
River	31.55	0.7891	31.55	0.7882	31.57	0.7885	31.60	0.7900	31.64	0.7905	31.61	0.7904	31.66	0.7918	**31.68**	**0.7922**
School	29.36	0.8044	29.41	0.8033	29.45	0.8041	29.51	0.8067	29.59	0.8091	29.50	0.8048	29.67	0.8123	**29.68**	**0.8124**
S-Residential	27.71	0.6728	27.79	0.6723	27.80	0.6725	27.85	0.6752	27.88	0.6758	27.80	0.6728	27.91	**0.6782**	**27.92**	0.6775
Square	30.84	0.8200	30.83	0.8181	30.87	0.8183	30.97	0.8218	31.06	0.8236	30.98	0.821	31.11	0.8256	**31.15**	**0.8266**
Stadium	29.63	0.8387	29.51	0.834	29.62	0.8358	29.74	0.8388	29.82	0.8413	29.76	0.8394	29.80	0.8420	**29.93**	**0.8439**
Storage Tanks	27.44	0.7664	27.50	0.7649	27.52	0.7648	27.58	0.7671	27.63	0.7692	27.58	0.767	27.66	0.7720	**27.68**	**0.7718**
Viaduct	28.99	0.7757	28.92	0.7711	28.96	0.7722	29.06	0.7759	29.14	0.7784	29.05	0.7753	29.16	0.7811	**29.25**	**0.7831**
Average	30.65	0.8086	30.66	0.8068	30.71	0.8076	30.77	0.8102	30.83	0.8114	30.78	0.8098	30.87	0.8138	**30.91**	**0.8142**

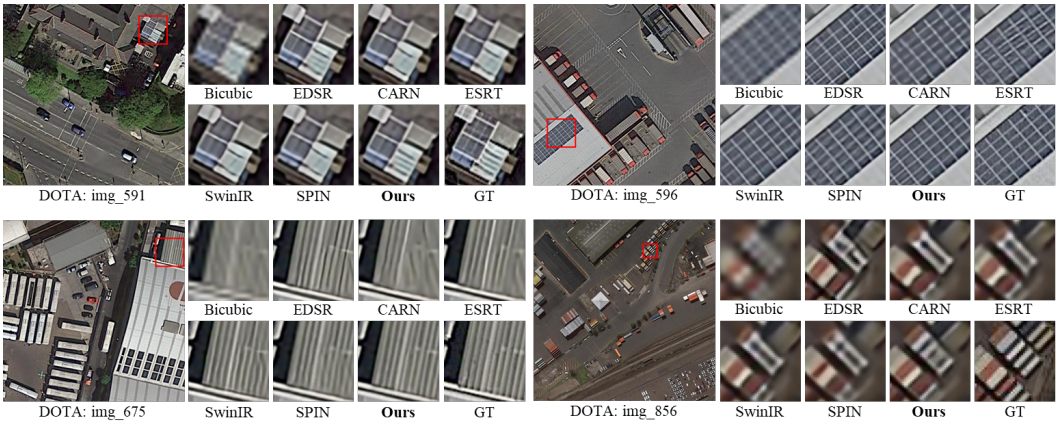


Figure 5: Qualitative comparison of SOTA efficient SR models for $\times 4$ SR task on DOTA dataset.

and $\beta_2 = 0.999$. Data augmentation includes random rotations of 90°, 180°, 270°, and horizontal flips on 64×64 patches. The channel number, embedding dimension of cross-attention, and MLP rate of small, medium, and final SpikeSR are set to $\{40, 24, 72\}$, $\{56, 24, 72\}$, and $\{64, 32, 100\}$, respectively. The number of SAGs is 4, with 2 SABs in each SAG. Time step is set to $\mathrm{T} = 4$.

Metrics. The widely used peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) are used to evaluate SR performance. The results are measured on the Y channel after converting RGB to YCbCr space. For a fair comparison, all SR methods are retained on an RTX 4090 GPU from scratch using the AID dataset, adhering to their official implementation settings.

3.1 Comparison with Efficient Models

We compare our SpikeSR with state-of-the-art (SOTA) efficient SR methods, including CNN-based models of CARN [1], LatticeNet [36], RLFN [23], etc, and transformer-based approaches of SwinIR

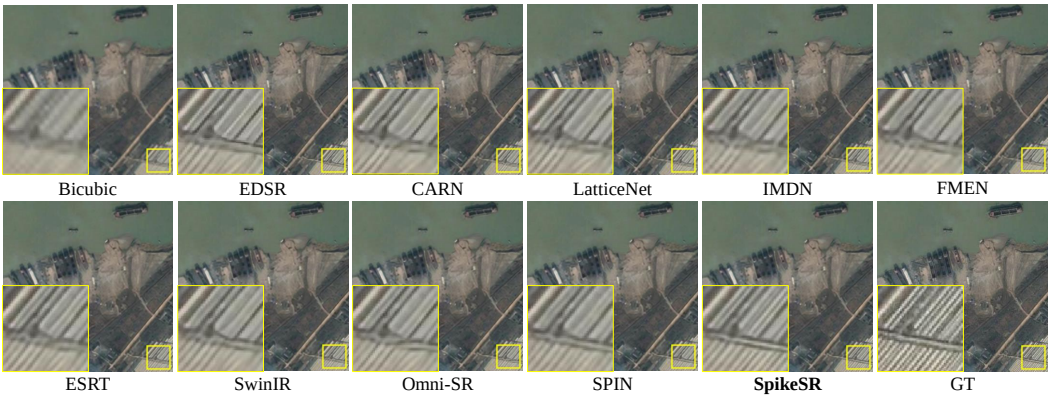


Figure 6: Qualitative comparison of SOTA efficient SR models for $\times 4$ SR task on DIOR dataset.

[31], SPIN [66], HiT-SR [68], *etc*. We also report results for the small and medium versions of our SpikeSR, denoted as SpikeSR-S and Spike-M, respectively.

Quantitative comparisons. The quantitative results of various methods are reported in Table 1. We can observe that our SpikeSR achieves the best performance across three benchmarks, outperforming SOTA CNN- and transformer-based SR models. For example, on AID, DOTA, and DIOR datasets, SpikeSR improves PSNR by 0.08 dB, 0.04 dB, and 0.1 dB, respectively, compared to the impressive SwinIR. Moreover, the small version of SpikeSR requires fewer parameters (472K vs. 897K) and FLOPs (15.21G vs. 23.56G) than SwinIR, yet achieves superior average performance.

Qualitative comparisons. Fig. 4, Fig. 5, and Fig. 6 present visual comparisons on AID, DOTA, and DIOR datasets, respectively. As shown in Fig. 4, SpikeSR effectively restores severely damaged textures, *e.g.*, the runway line in the playground. By contrast, other SR models fail to recover such weak high-frequency details. In Fig. 5, the reconstruction of "img_591" highlights that SpikeSR produces results closest to the GT, while other methods like SPIN recovers unrealistic results. Moreover, Fig. 6 further demonstrates that SpikeSR consistently delivers superior visual quality, restoring more textural information compared to large-capacity ANN-based model like Omni-SR.

4 Ablation Study

We conduct ablation studies to assess the impact of key components in SpikeSR. The experimental results in Table 3 are measured on the AID-tiny dataset [58]. In particular, the Baseline model is constructed by removing the HDA and DSA and replacing them with standard temporal attention (TA), channel attention (CA), and spatial attention (SA) mechanisms. For a fair comparison, we increase the number of m, n, and channel dimensions to 8, 4, and 256, respectively, which ensures a similar number of parameters with our SpikeSR. Similarly, those settings of Variant-A are modified to 10, 5, and 128, respectively.

Effectiveness of HDA and DSA. Table 3 indicates how the SR performance is influenced by the HDA and DSA. Comparing the PSNR values of the Baseline and Variant-A reveals that HDA contributes a 0.12 dB improvement. This suggests that enhancing the correlation between the temporal and channel dimensions delivers better recalibration of the membrane potentials. More intuitively, we visualize the feature maps to highlight the impact of HDA, as shown in Fig. 7(b). The results illustrate that HDA effectively refines the feature representation by emphasizing salient details and suppressing irrelevant information.

By introducing the proposed DSA to the Baseline model, the PSNR results can be improved by a large margin of 0.21 dB. When employing HDA and DSA simultaneously, the resulting SpikeSR achieves an additional 0.08 dB improvement compared to Variant-B. To better demonstrate its effectiveness in capturing global self-similarity priors, we provide the LAM visualization and diffusion index in Fig. 7(a). As observed, DSA generates more pronounced LAMs and significantly increases the DI, indicating that the model activates more valuable pixels for accurate SR.

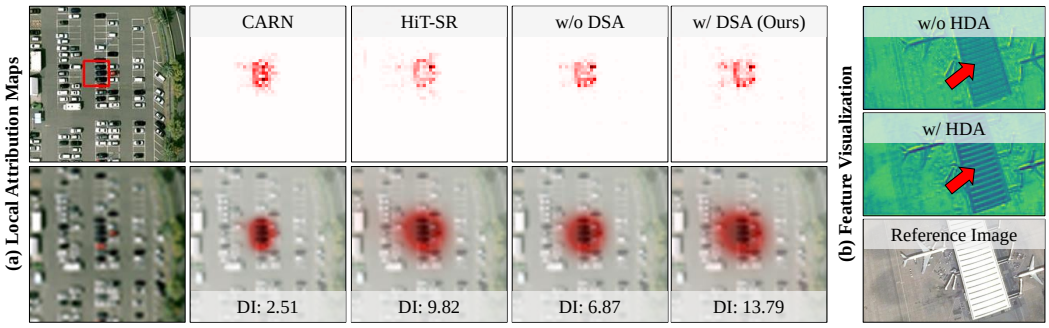


Figure 7: (a) Analysis of local attribution maps (LAMs) [16] and diffusion index (DI). The proposed DAS helps SpikeSR exploit more useful information against CARN and HiT-SR. (b) Feature visualizations. The feature obtained by HDA is sharper and preserves more details, indicating high-quality feature representations.

Table 3: Ablation on different variants of our SpikeSR.

Methods	TA	CA	SA	HDA	DSA	#Param.	PSNR (dB)
Baseline	✓	✓	✓			1120K	27.80
Variant-A			✓	✓		1062K	27.92
Variant-B	✓	✓			✓	1009K	28.11
SpikeSR				✓	✓	1042K	**28.19**

Table 4: Ablation on feature pyramid and deformable convolution.

Methods	w/o Pyramid	w/o DConv	DSA (Full)
#Param.	1042K	918K	1042K
FLOPs	31.41G	28.86G	33.00G
PSNR (dB)	28.14	28.07	**28.19**

Table 5: Performance and complexity analysis of SpikeSR with different numbers of blocks.

Blocks	2	3	4	5	6
#Param.	558K	800K	1042K	1284K	1526K
FLOPs	17.47G	25.26G	33.00G	40.84G	48.60G
PSNR (dB)	28.11	28.17	**28.19**	28.16	**28.20**

Feature Pyramid and DConv. Due to the scale diversity of objects in RSIs, we constructed a feature pyramid to grasp the self-similarity in multiple levels. As listed in Table 4, the use of feature pyramid improves the performance by 0.05 dB without introducing additional parameters. Furthermore, to demonstrate the effectiveness of deformable convolution, we remove this component, which leads to a severe performance drop by 0.12 dB. This illustrates that the self-similar patches contain massive irrelevant and misaligned contents, and direct fusion may introduce interference, thus resulting in suboptimal performance.

Network Depth. We evaluate the impact of the network depth by changing the number of SAGs of our SpikeSR from 2 to 6 blocks. As reported in Table 5, SpikeSR achieves the highest SR performance when $m = 3$. While increasing the number of m may further improve the reconstruction, it also brings larger model size. Therefore, we set $m = 4$ in our final model, considering the trade-off between performance and computational complexity.

5 Conclusion and Limitation

In this paper, we investigate the application of SNNs for efficient SR of remote sensing images. Motivated by the observation that LIF neurons exhibit a higher spike rate in degraded images, we integrate SNNs with convolutions for improved feature representation. Besides, a hybrid dimension attention is employed to modulate the spike response, further refining salient information. To incorporate valuable prior knowledge for more accurate SR, we propose a deformable similarity attention module, capturing global self-similarity across multiple feature levels. Extensive experiments on various remote sensing datasets demonstrate the efficacy and effectiveness of the proposed SpikeSR model. We believe our exploration can facilitate the practical application of energy-efficient models in remote sensing area.

SpikeSR still has some limitations. First, its representational capacity remains improvable. Second, the reliance on attention mechanisms introduces additional computational overhead, limiting efficiency. We leave these issues for future exploration.

Acknowledgment. This work is supported in part by the National Natural Science Foundation of China (423B2104, 623B2080), and in part by the Natural Science Foundation of Heilongjiang Province of China for Excellent Youth Project (YQ2024F006), and in part by the Open Research Fund from Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ) (GML-KF-24-09).

References

[1] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *ECCV*, pages 252–268, 2018.

[2] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *CVPR*, pages 12299–12310, 2021.

[3] Xiangyu Chen, Xintao Wang, Jiantao Zhou, Yu Qiao, and Chao Dong. Activating more pixels in image super-resolution transformer. In *CVPR*, pages 22367–22377, 2023.

[4] Zheng Chen, Yulun Zhang, Jinjin Gu, Linghe Kong, Xiaokang Yang, and Fisher Yu. Dual aggregation transformer for image super-resolution. In *ICCV*, pages 12312–12321, 2023.

[5] Haram Choi, Jeongmin Lee, and Jihoon Yang. N-gram in swin transformers for efficient lightweight image super-resolution. In *CVPR*, pages 2071–2081, 2023.

[6] Julien Cornebise, Ivan Oršolić, and Freddie Kalaitzis. Open high-resolution satellite imagery: The worldstrat dataset–with application to super-resolution. *NeurIPS*, 35:25979–25991, 2022.

[7] Renwei Dian, Leyuan Fang, and Shutao Li. Hyperspectral image super-resolution via non-local sparse tensor factorization. In *CVPR*, pages 5344–5353, 2017.

[8] Peter U. Diehl, Daniel Neil, Jonathan Binas, Matthew Cook, Shih-Chii Liu, and Michael Pfeiffer. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *IJCNN*, pages 1–8, 2015.

[9] Jianhao Ding, Zhaofei Yu, Yonghong Tian, and Tiejun Huang. Optimal ann-snn conversion for fast and accurate inference in deep spiking neural networks. In *IJCAI*, pages 2328–2336, 2021.

[10] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE TPAMI*, 38(2):295–307, 2016.

[11] Runmin Dong, Shuai Yuan, Bin Luo, Mengxuan Chen, Jinxiao Zhang, Lixian Zhang, Weijia Li, Juepeng Zheng, and Haohuan Fu. Building bridges across spatial and temporal resolutions: Reference-based super-resolution via change priors and conditional diffusion model. In *CVPR*, pages 27684–27694, 2024.

[12] Zongcai Du, Ding Liu, Jie Liu, Jie Tang, Gangshan Wu, and Lean Fu. Fast and memory-efficient network towards efficient image super-resolution. In *CVPRW*, pages 853–862, 2022.

[13] Jinsheng Fang, Hanjiang Lin, Xinyu Chen, and Kun Zeng. A hybrid network of cnn and transformer for lightweight image super-resolution. In *CVPRW*, pages 1103–1112, 2022.

[14] Wei Fang, Yanqi Chen, Jianhao Ding, Zhaofei Yu, Timothée Masquelier, Ding Chen, Liwei Huang, Huihui Zhou, Guoqi Li, and Yonghong Tian. Spikingjelly: An open-source machine learning infrastructure platform for spike-based intelligence. *Science Advances*, 9(40):eadi1480, 2023.

[15] Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. *NeurIPS*, 34:21056–21069, 2021.

[16] Jinjin Gu and Chao Dong. Interpreting super-resolution networks with local attribution maps. In *CVPR*, pages 9199–9208, 2021.

[17] Hang Guo, Yong Guo, Yaohua Zha, Yulun Zhang, Wenbo Li, Tao Dai, Shu-Tao Xia, and Yawei Li. Mambairv2: Attentive state space restoration. In *CVPR*, pages 28124–28133, 2025.

[18] Hang Guo, Jinmin Li, Tao Dai, Zhihao Ouyang, Xudong Ren, and Shu-Tao Xia. Mambair: A simple baseline for image restoration with state-space model. In *ECCV*, pages 222–241, 2024.

[19] Yifan Hu, Lei Deng, Yujie Wu, Man Yao, and Guoqi Li. Advancing spiking neural networks toward deep residual learning. *IEEE TNNLS*, 36(2):2353–2367, 2025.

[20] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. Lightweight image super-resolution with information multi-distillation network. In *ACM MM*, pages 2024–2032, 2019.

[21] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, pages 1646–1654, 2016.

[22] Seijoon Kim, Seongsik Park, Byunggook Na, and Sungroh Yoon. Spiking-yolo: spiking neural network for energy-efficient object detection. In *AAAI*, pages 11270–11277, 2020.

[23] Fangyuan Kong, Mingxi Li, Songwei Liu, Ding Liu, Jingwen He, Yang Bai, Fangmin Chen, and Lean Fu. Residual local feature network for efficient super-resolution. In *CVPRW*, pages 766–776, 2022.

[24] Souvik Kundu, Gourav Datta, Massoud Pedram, and Peter A Beerel. Spike-thrift: Towards energy-efficient deep spiking neural networks by limiting spiking activity via attention-guided compression. In *WACV*, pages 3953–3962, 2021.

[25] Yuxiang Lan, Yachao Zhang, Xu Ma, Yanyun Qu, and Yun Fu. Efficient converted spiking neural network for 3d and 2d classification. In *ICCV*, pages 9211–9220, 2023.

[26] Chankyu Lee, Adarsh Kumar Kosta, Alex Zihao Zhu, Kenneth Chaney, Kostas Daniilidis, and Kaushik Roy. Spike-flownet: event-based optical flow estimation with energy-efficient hybrid neural networks. In *ECCV*, pages 366–382, 2020.

[27] Jun Haeng Lee, Tobi Delbruck, and Michael Pfeiffer. Training deep spiking neural networks using backpropagation. *Frontiers in neuroscience*, 10:508, 2016.

[28] Sen Lei and Zhenwei Shi. Hybrid-scale self-similarity exploitation for remote sensing image super-resolution. *IEEE TGRS*, 60:1–10, 2021.

[29] Ke Li, Gang Wan, Gong Cheng, Liqiu Meng, and Junwei Han. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS J P&RS*, 159:296–307, 2020.

[30] Yuhang Li, Shikuang Deng, Xin Dong, and Shi Gu. Error-aware conversion from ann to snn via post-training parameter calibration. *IJCV*, 132:3586–3609, 2024.

[31] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *ICCV*, pages 1833–1844, 2021.

[32] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, pages 136–144, 2017.

[33] Jie Liu, Chao Chen, Jie Tang, and Gangshan Wu. From coarse to fine: Hierarchical pixel integration for lightweight image super-resolution. In *AAAI*, volume 37, pages 1666–1674, 2023.

[34] Jie Liu, Jie Tang, and Gangshan Wu. Residual feature distillation network for lightweight image super-resolution. In *ECCV*, pages 41–55, 2020.

[35] Zhisheng Lu, Juncheng Li, Hong Liu, Chaoyan Huang, Linlin Zhang, and Tieyong Zeng. Transformer for single image super-resolution. In *CVPRW*, pages 457–466, 2022.

[36] Xiaotong Luo, Yuan Xie, Yulun Zhang, Yanyun Qu, Cuihua Li, and Yun Fu. Latticenet: Towards lightweight image super-resolution with lattice block. In *ECCV*, pages 272–289, 2020.

[37] Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997.

[38] Yiqun Mei, Yuchen Fan, and Yuqian Zhou. Image super-resolution with non-local sparse attention. In *CVPR*, pages 3517–3526, 2021.

[39] Ben Niu, Weilei Wen, Wenqi Ren, Xiangde Zhang, Lianping Yang, Shuzhen Wang, Kaihao Zhang, Xiaochun Cao, and Haifeng Shen. Single image super-resolution via a holistic attention network. In *ECCV*, pages 191–207, 2020.

[40] Federico Paredes-Vallés, Kirk YW Scheper, and Guido CHE De Croon. Unsupervised learning of a hierarchical spiking neural network for optical flow estimation: From events to global motion perception. *IEEE TPAMI*, 42(8):2051–2064, 2019.

[41] Xuerui Qiu, Rui-Jie Zhu, Yuhong Chou, Zhaorui Wang, Liang-jian Deng, and Guoqi Li. Gated attention coding for training high-performance and efficient spiking neural networks. In *AAAI*, pages 601–610, 2024.

[42] Muhammed T Razzak, Gonzalo Mateo-García, Gurvan Lecuyer, Luis Gómez-Chova, Yarin Gal, and Freddie Kalaitzis. Multi-spectral multi-image super-resolution of sentinel-2 with radiometric consistency losses and its effect on building delineation. *ISPRS J P&RS*, 195:1–13, 2023.

[43] 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*, pages 1874–1883, 2016.

[44] Xinyu Shi, Zecheng Hao, and Zhaofei Yu. Spikingresformer: Bridging resnet and vision transformer in spiking neural networks. In *CVPR*, pages 5610–5619, June 2024.

[45] Tianyu Song, Guiyue Jin, Pengpeng Li, Kui Jiang, Xiang Chen, and Jiyu Jin. Learning a spiking neural network for efficient image deraining. *arXiv preprint arXiv:2405.06277*, 2024.

[46] Christoph Stöckl and Wolfgang Maass. Optimized spiking neurons can classify images with high accuracy through temporal coding with two spikes. *Nature Machine Intelligence*, 3(3):230–238, 2021.

[47] Jian-Nan Su, Guodong Fan, Min Gan, Guang-Yong Chen, Wenzhong Guo, and C. L. Philip Chen. Revealing the dark side of non-local attention in single image super-resolution. *IEEE TPAMI*, 46(12):11476–11490, 2024.

[48] Jian Sun, Zongben Xu, and Heung-Yeung Shum. Image super-resolution using gradient profile prior. In *CVPR*, pages 1–8, 2008.

[49] Long Sun, Jiangxin Dong, Jinhui Tang, and Jinshan Pan. Spatially-adaptive feature modulation for efficient image super-resolution. In *ICCV*, pages 13190–13199, 2023.

[50] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *CVPR*, pages 3147–3155, 2017.

[51] Hang Wang, Xuanhong Chen, Bingbing Ni, Yutian Liu, and Jinfan Liu. Omni aggregation networks for lightweight image super-resolution. In *CVPR*, pages 22378–22387, 2023.

[52] Longguang Wang, Xiaoyu Dong, Yingqian Wang, Xinyi Ying, Zaiping Lin, Wei An, and Yulan Guo. Exploring sparsity in image super-resolution for efficient inference. In *CVPR*, pages 4917–4926, 2021.

[53] Zhihao Wang, Jian Chen, and Steven CH Hoi. Deep learning for image super-resolution: A survey. *IEEE TPAMI*, 43(10):3365–3387, 2020.

[54] Ziqing Wang, Yuetong Fang, Jiahang Cao, Qiang Zhang, Zhongrui Wang, and Renjing Xu. Masked spiking transformer. In *ICCV*, pages 1761–1771, 2023.

[55] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *CVPR*, pages 3974–3983, 2018.

[56] Gui-Song Xia, Jingwen Hu, Fan Hu, Baoguang Shi, Xiang Bai, Yanfei Zhong, Liangpei Zhang, and Xiaoqiang Lu. Aid: A benchmark data set for performance evaluation of aerial scene classification. *IEEE TGRS*, 55(7):3965–3981, 2017.

[57] Yi Xiao, Qiangqiang Yuan, Kui Jiang, Yuzeng Chen, Qiang Zhang, and Chia-Wen Lin. Frequency-assisted mamba for remote sensing image super-resolution. *IEEE TMM*, 27:1783–1796, 2025.

[58] Yi Xiao, Qiangqiang Yuan, Kui Jiang, Jiang He, Chia-Wen Lin, and Liangpei Zhang. Ttst: A top-k token selective transformer for remote sensing image super-resolution. *IEEE TIP*, 33:738–752, 2024.

[59] Zheyu Yang, Yujie Wu, Guanrui Wang, Yukuan Yang, Guoqi Li, Lei Deng, Jun Zhu, and Luping Shi. Dashnet: A hybrid artificial and spiking neural network for high-speed object tracking. *arXiv preprint arXiv:1909.12942*, 2019.

[60] Jing Yao, Danfeng Hong, Jocelyn Chanussot, Deyu Meng, Xiaoxiang Zhu, and Zongben Xu. Cross-attention in coupled unmixing nets for unsupervised hyperspectral super-resolution. In *ECCV*, pages 208–224, 2020.

[61] Man Yao, Huanhuan Gao, Guangshe Zhao, Dingheng Wang, Yihan Lin, Zhaoxu Yang, and Guoqi Li. Temporal-wise attention spiking neural networks for event streams classification. In *CVPR*, pages 10221–10230, 2021.

[62] Man Yao, JiaKui Hu, Tianxiang Hu, Yifan Xu, Zhaokun Zhou, Yonghong Tian, Bo XU, and Guoqi Li. Spike-driven transformer v2: Meta spiking neural network architecture inspiring the design of next-generation neuromorphic chips. In *ICLR*.

[63] Man Yao, Xuerui Qiu, Tianxiang Hu, Jiakui Hu, Yuhong Chou, Keyu Tian, Jianxing Liao, Luziwei Leng, Bo Xu, and Guoqi Li. Scaling spike-driven transformer with efficient spike firing approximation training. *IEEE TPAMI*, 47(4):2973–2990, 2025.

[64] Man Yao, Ole Richter, Guangshe Zhao, Ning Qiao, Yannan Xing, Dingheng Wang, Tianxiang Hu, Wei Fang, Tugba Demirci, Michele De Marchi, et al. Spike-based dynamic computing with asynchronous sensing-computing neuromorphic chip. *Nature Communications*, 15(1):4464, 2024.

[65] Man Yao, Guangshe Zhao, Hengyu Zhang, Yifan Hu, Lei Deng, Yonghong Tian, Bo Xu, and Guoqi Li. Attention spiking neural networks. *IEEE TPAMI*, 45(8):9393–9410, 2023.

[66] Aiping Zhang, Wenqi Ren, Yi Liu, and Xiaochun Cao. Lightweight image super-resolution with superpixel token interaction. In *ICCV*, pages 12728–12737, 2023.

[67] Kaibing Zhang, Xinbo Gao, Dacheng Tao, and Xuelong Li. Single image super-resolution with non-local means and steering kernel regression. *IEEE TIP*, 21(11):4544–4556, 2012.

[68] Xiang Zhang, Yulun Zhang, and Fisher Yu. Hit-sr: Hierarchical transformer for efficient image super-resolution. In *ECCV*, pages 483–500, 2024.

[69] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, pages 286–301, 2018.

[70] Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li. Going deeper with directly-trained larger spiking neural networks. In *AAAI*, pages 11062–11070, 2021.

[71] Man Zhou, Keyu Yan, Jinshan Pan, Wenqi Ren, Qi Xie, and Xiangyong Cao. Memory-augmented deep unfolding network for guided image super-resolution. *IJCV*, 131(1):215–242, 2023.

[72] Rui-Jie Zhu, Malu Zhang, Qihang Zhao, Haoyu Deng, Yule Duan, and Liang-Jian Deng. Tcja-snn: Temporal-channel joint attention for spiking neural networks. *IEEE TNNLS*, pages 1–14, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the main limitation of representation capability and model efficiency in the last section (Conclusion and Limitation) of the main paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The paper provides a cross-reference to the Gumbel-Softmax formulation, as detailed in Section 2.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide comprehensive details of the model architecture, together with training details to reproduce the main experimental results, supporting the main claims and conclusions.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The adopted remote sensing datasets are publicly available. The datasets used in this work are properly described and cited in Section 3, and will be released alongside our code if the paper is accepted.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All the details in this regard are provided in Section 3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We do not report error bars.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Our SpikeSR is trained on a single NVIDIA RTX 4090 GPU, as detailed in Section 3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read and understood the NeurIPS Code of Ethics and the associated policies, and we believe that the research in the paper fully conforms to the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: There is no negative societal impact associated with our paper. Our work focuses on deep-learning-based methods for potential remote sensing applications, which is discussed in the conclusion of the main paper.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA] .

    Justification: We are aware of no potential risk for any misuse of our work.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have properly cited the open datasets in Section 3. Their licenses permit use within academic scope.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We do not release any new assets, except for our codes, which are made publicly accessible alongside documentation.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing. The human data are acquired from public datasets, properly cited in the manuscript.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: This paper does not include LLMs as any important, original, or non-standard components of the core methods.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.