SPIKEINPAINT: HYBRID ANN-SNN IMAGE INPAINTING WITH TERNARY SPIKE CODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Image inpainting is dominated by generators that perform dense floating-point computation throughout the network, incurring substantial computational and energy costs. We investigate whether sparse spiking computation can match the reconstruction quality of dense computation at a lower activation cost. We present SpikeInpaint, a hybrid ANN-SNN generator that places spiking computation only where the spatial resolution is lowest: the four FFC blocks of the bottleneck and the two channel-transition modules on either side of it. The bottleneck combines ternary hard-reset spiking neurons with channel compression, and a channel-wise gate in the decoder fuses the ANN and SNN paths per channel. Experiments on Places2 and CelebA-HQ show that SpikeInpaint achieves 95.89% spike sparsity during inference, within 0.09 dB of an otherwise identical all-ANN generator on Places2. These results show that sparse spiking computation can support dense image generation, extending SNNs to dense generative vision.

1 INTRODUCTION

Image inpainting aims to reconstruct missing regions in corrupted images. Existing methods have achieved remarkable performance using encoder-decoder architectures (Pathak et al., 2016), mask-adaptive convolutions (Yu et al., 2019), attention mechanisms (Yu et al., 2018; Li et al., 2022), and diffusion models (Lugmayr et al., 2022; Zhang et al., 2023; Yang et al., 2023; Kim et al., 2025). Despite their architectural differences, these artificial neural network (ANN) approaches rely almost entirely on dense floating-point computation throughout the generator. In contrast, spiking neural networks (SNNs) perform sparse spike-based computation, activating neurons only when membrane potentials exceed firing thresholds (Maass, 1997; Roy et al., 2019). This property has enabled competitive performance in image classification (Hu et al., 2021; Zheng et al., 2021) and object detection (Kim et al., 2020; Luo et al., 2024; Miao et al., 2025b), but it has not been applied to masked image inpainting under an adversarial objective. Spiking computation is also not a drop-in replacement: thresholding quantizes every feature that passes through it, and the number of spike events grows with the channel count at that location.

Inspired by recent advances in hybrid ANN-SNN vision models, we investigate whether sparse spiking computation can also benefit image inpainting under the GAN framework (Goodfellow et al., 2014), whose encoder-decoder architecture is widely adopted for image inpainting (Pathak et al., 2016; Nazeri et al., 2019; Liu et al., 2021) and is easy to equip with SNN modules. Since image inpainting relies heavily on long-range contextual information, we adopt Fast Fourier Convolutions (FFCs) (Chi et al., 2020; Suvorov et al., 2022), which naturally combine local spatial features with global Fourier representations, providing a suitable backbone for hybrid ANN-SNN image inpainting.

We propose SpikeInpaint, a hybrid ANN-SNN generator that builds upon the FFC backbone and integrates sparse spiking computation into the image inpainting pipeline. The generator retains conventional ANN computation in the encoder and decoder while introducing spiking computation into the Fourier bottleneck and the surrounding channel-transition modules.

Our contributions are:

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

- We propose SpikeInpaint, a hybrid ANN–SNN inpainting generator that localizes spiking computation to the FFC bottleneck and the two adjacent channel-transition modules, leaving the encoder and the remaining decoder stages dense.
- We introduce TriHardLIF, a ternary hard-reset neuron that makes training and single-timestep inference dynamics identical, together with SNN Bottleneck, which compresses channels before spike generation, and SNNGate Light, which fuses the ANN and SNN paths per channel during decoding.
- We show that the bottleneck fires on a small fraction of spiking neurons while reconstruction quality stays within 0.09 dB of an all-ANN generator of the same architecture on Places2.

2 Related Work

2.1 Image Inpainting

Deep image inpainting has progressed from convolution-based architectures to global context modelling and large-scale generative priors. Early encoder-decoder networks (Pathak et al., 2016) are subsequently improved by mask-aware operators, such as partial convolution (Liu et al., 2018) and gated convolution (Yu et al., 2019), which restrict feature aggregation to valid regions and learn adaptive feature selection. Although these methods are effective, they remain limited by the locality of stacked convolutions. Attention-based approaches alleviate this limitation by explicitly modelling long-range dependencies through contextual attention (Yu et al., 2018) or mask-aware Transformers (Li et al., 2022). Two-stage methods further separate structure prediction from texture synthesis (Nazeri et al., 2019), improving reconstruction quality with extra edge completion. Frequency-domain approaches instead enlarge the receptive field through spectral operations. Fast Fourier Convolutions (FFCs) (Chi et al., 2020) and the FFC-based LaMa (Suvorov et al., 2022) demonstrate that global Fourier representations can effectively handle large irregular masks without explicit attention. More recently, diffusion-based methods (Lugmayr et al., 2022; Zhu et al., 2024) achieve high-fidelity reconstruction through iterative denoising, but typically require hundreds of forward passes per image. Despite these architectural advances, existing inpainting methods still rely predominantly on dense floating-point computation throughout the generation process.

2.2 Spiking Neural Networks

SNNs perform sparse spike-based computation by activating neurons only when membrane potentials exceed predefined thresholds (Maass, 1997). Their non-differentiable spike generation is commonly optimized using surrogate gradients (Neftci et al., 2019; Wu et al., 2018), enabling deep SNNs to achieve competitive performance on image classification (Hu et al., 2021; Zheng et al., 2021) and object detection (Kim et al., 2020; Luo et al., 2024). Hybrid ANN-SNN designs incorporating residual connections and attention have since pushed discriminative accuracy further while reducing computation (Miao et al., 2025a;b). Spiking computation has been used for image generation before: FSVAE generates images with an entirely spiking variational autoencoder (Kamata et al., 2022), showing that spiking networks are not limited to discriminative tasks. In inpainting, the closest prior work is a shallow STDP texture model for texture completion (Kumar et al., 2019), with neither a deep backbone nor adversarial training. To our knowledge, no published generator performs end-to-end inpainting with spiking neurons trained by surrogate gradients under adversarial objectives.

3 Methods

3.1 Network Architecture

Figure 1 illustrates the overall architecture of SpikeInpaint. The proposed generator follows a symmetric encoder-decoder design built upon the FFC backbone, combining conventional ANN computation with sparse spiking computation in a unified framework. The encoder and decoder preserve standard convolutional operations, while dedicated SNN modules are introduced around the Fourier

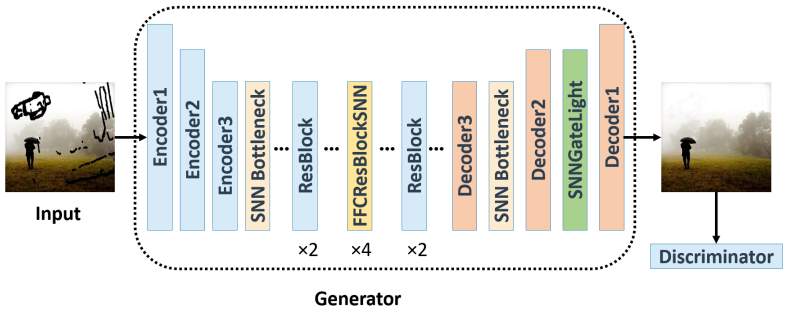
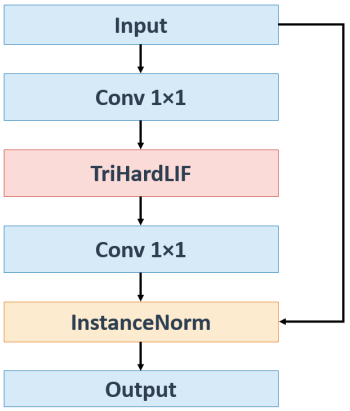
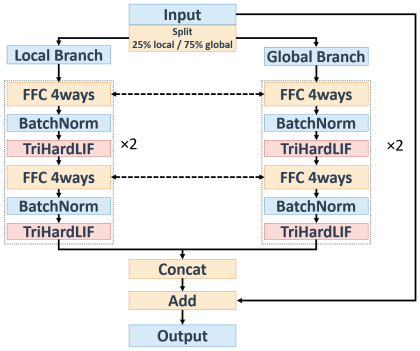


Figure 1: Generator architecture of SpikeInpaint. FFC ResBlock-SNN modules apply TriHardLIF for spike-wise computations; SNN Bottleneck and SNNGate Light modules embed spiking neurons at channel-density transitions. All other blocks retain standard ANN activations.



(a) The structure of the SNN Bottleneck. This module is applied at the end of the encoder and inside the first decoder layer, maintaining a symmetric structure in the network.



(b) FFC ResBlock-SNN. Two FFC-BN-TriHardLIF triplets operate on split local and global branches with a residual connection. Four such blocks form the bottleneck core.

Figure 2: The two spiking building blocks of the bottleneck. (a) SNN Bottleneck compresses feature channels before spike generation, applied at the encoder-to-bottleneck and bottleneck-to-decoder transitions. (b) FFC ResBlock-SNN replaces the activation in both FFC branches with TriHardLIF; four such blocks form the bottleneck core.

bottleneck, where feature representations become increasingly compact and global context is aggregated. The bottleneck consists of four FFC ResBlock-SNN modules between two residual blocks, and the decoder recovers the spatial resolution with transposed convolutions while SNNGate Light fuses the ANN and SNN paths.

3.2 TriHardLIF: Ternary Hard-Reset LIF

Conventional Leaky Integrate-and-Fire (LIF) neurons adopt a soft-reset mechanism that subtracts the firing threshold from the membrane potential after each spike, preserving residual states for multi-step temporal modelling. However, the generator is trained and evaluated under a single-timestep setting to maintain the same inference latency as ANN-based image inpainting. Under this regime, soft-reset dynamics introduce additional membrane-state dependencies during optimization that are absent during inference. These states never exist at inference because each forward pass starts from zero membrane potential. This difference is particularly undesirable for dense image reconstruction, where every feature directly contributes to the final output.

To address this issue, we propose TriHardLIF, a ternary hard-reset LIF neuron. Instead of subtracting the threshold after firing, TriHardLIF resets the membrane potential to zero, ensuring identical neuron dynamics during training and inference. The neuron produces ternary spike responses $\{-1, 0, +1\}$ while remaining compatible with surrogate-gradient optimization.

Each forward pass consists of membrane integration, ternary spike generation, and hard reset. The membrane potential evolves as:

$$v[t] = \alpha \cdot v[t-1] + x[t], \quad \alpha = e^{-1/\tau}, \tag{1}$$

where τ controls the membrane decay. Spike generation is defined as:

$$s[t] = \mathrm{sgn}(v[t]) \cdot \mathbf{1}\{|v[t]| > \theta\}, \tag{2}$$

yielding ternary outputs $\{-1, 0, +1\}$. After firing, active neurons are reset according to:

$$v[t] \leftarrow v[t] \cdot \mathbf{1}\{|s[t]| < 0.5\}. \tag{3}$$

Since the indicator function in Eq. 2 is non-differentiable, gradients are approximated using the arctan surrogate function (Neftci et al., 2019):

$$\frac{\partial s}{\partial v} \approx \sigma'(v-\theta) + \sigma'(-v-\theta), \qquad \sigma'(x) = \frac{1}{1+(\pi x)^2}, \tag{4}$$

while the forward pass retains exact ternary spike generation through the straight-through estimator (Bengio et al., 2013). Membrane states are reset after every iteration to prevent state accumulation across training batches and inference samples.

3.3 SNN Bottleneck

Spiking computation is introduced at the two channel-dense transitions between the encoder, bottleneck, and decoder. Directly applying spiking neurons at these locations would generate a large number of spike events because the feature dimensionality reaches its maximum in the deepest layer of the encoder and bottleneck. To reduce the number of spikes, we introduce SNN Bottleneck, which compresses feature channels before spike generation and restores the original dimensionality afterward. Spiking computation is, therefore, carried out in a compact latent space, while the residual connection preserves the surrounding ANN feature representation.

One SNN Bottleneck is placed at the end of the encoder and one at the beginning of the decoder, where the features are spatially compact and already encode high-level semantics.

Each module is defined as:

$$\text{SNNBottleneck}(x) = \text{IN}(x + W_{\text{exp}} \cdot \text{TriHardLIF}(W_{\text{cmp}} \cdot x)), \tag{5}$$

where W_{cmp} projects the input feature into a lower-dimensional space and W_{exp} restores the original channel dimension. The ANN branch preserves dense feature information, whereas the spiking branch contributes a sparse residual correction. The compressed representation increases activation selectivity, so only the most informative responses cross the threshold.

3.4 FFC ResBlock-SNN

Long-range context is essential for image inpainting, particularly when reconstructing large missing regions. We therefore adopt the FFC block of Chi et al. (2020) as the bottleneck backbone, without modification. We do not claim that its spectral formulation is what makes it effective for inpainting: Chu et al. (2023) attribute the benefit of this design largely to its large receptive field rather than to the transform. We use it because its local and global branches carry features with different statistics, which makes the placement question well posed.

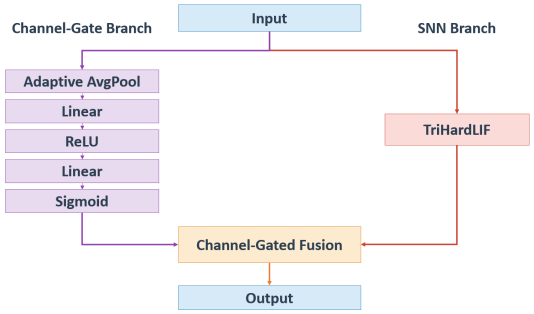


Figure 3: SNNGate Light. A per-channel gate blends full-precision ANN features with ternary SNN features during upsampling.

To introduce sparse spiking computation into this representation, we replace the activation in each FFC branch with TriHardLIF, yielding the proposed FFC ResBlock-SNN. Each residual block consists of two consecutive FFC-BN-TriHardLIF units. Batch normalization and TriHardLIF are applied independently to the local and global branches after every FFC layer. Both branches share the same neuron parameters (τ, θ) and differ only in the statistics of their inputs, which lets us read branch-wise firing-rate differences as a property of the representation rather than of the neuron configuration.

The FFC ResBlock-SNN is formulated as:

$$\begin{aligned}(x_l, x_g) &= \text{Split}_\gamma(x),\\ (x_l', x_g') &= \mathcal{F}_{\text{FBT}}^{(2)}\Big(\mathcal{F}_{\text{FBT}}^{(1)}(x_l, x_g)\Big),\\ \text{Out} &= x + \text{Concat}(x_l', x_g'),\end{aligned} \tag{6}$$

where split_γ partitions the input channels into local and global branches, and concat restores the original feature layout. $\mathcal{F}_{\text{FBT}}$ denotes an FFC-BN-TriHardLIF unit applied to both branches. Four FFC ResBlock-SNN modules form the core of the bottleneck. As shown in the experimental analysis, the two branches indeed exhibit distinct firing statistics.

3.5 SNNGate Light: Channel-Gated ANN-SNN Fusion

Decoder channels differ in how much ternary quantization they tolerate, so a single fixed precision is suboptimal: dense regions such as background need the full-precision ANN path, while others can be represented sparsely. SNNGate Light uses a per-channel gate computed from a globally pooled descriptor, and is applied in the second decoder stage. A lightweight bottleneck MLP predicts channel-wise gates. Specifically, global average pooling (GAP) first aggregates the spatial dimensions of the feature map, producing a channel descriptor. The gate produces a soft weight that determines the contribution of the full-precision ANN branch and the ternary SNN branch for each channel.

The gating fusion is defined as:

$$\begin{aligned}g &= \sigma(W_2 \cdot \text{ReLU}(W_1 \cdot \text{GAP}(x))) \in (0,1)^C,\\ y &= x \odot (1-g) + \text{TriHardLIF}(x) \odot g,\end{aligned} \tag{7}$$

where W_1 and W_2 denote the channel reduction and expansion projections, respectively; and $GAP(\cdot)$ denotes global average pooling. The gating parameters are optimized jointly with the reconstruction objective without additional regularization. Unlike Gated Convolution (Yu et al., 2019), which predicts spatial masks for valid-region selection, our module estimates channel-wise weights to adaptively balance full-precision and ternary representations.

3.6 Training Losses

The generator is trained with a reconstruction, an adversarial, a perceptual, and a style loss. The reconstruction loss is defined as:

$$\mathcal{L}_{\ell_1} = \|(1 - M) \odot (\hat{I} - I_{\text{gt}})\|_1, \tag{8}$$

where M denotes the valid-region mask and $\hat{I}$ denotes the generated image.

The adversarial objective adopts the non-saturating GAN formulation (Goodfellow et al., 2014):

$$\mathcal{L}_{\text{adv}} = -\mathbb{E}_{\hat{I}}\big[\log D(\hat{I})\big], \tag{9}$$

where the discriminator is implemented as a five-layer PatchGAN (Isola et al., 2017) with spectral normalization (Miyato et al., 2018).

The perceptual and style losses follow (Johnson et al., 2016; Gatys et al., 2016):

$$\mathcal{L}_{\text{perc}} = \sum_l \big\|\phi_l(\hat{I}) - \phi_l(I_{\text{gt}})\big\|_1. \tag{10}$$

$$\mathcal{L}_{\text{style}} = \sum_l \big\|G(\phi_l(\hat{I})) - G(\phi_l(I_{\text{gt}}))\big\|_1, \tag{11}$$

where $G(F) = FF^\top/(HWC)$ denotes the Gram matrix. The overall objective is:

$$\begin{aligned}\mathcal{L}_G = \lambda_{\ell_1}\mathcal{L}_{\ell_1} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}} + \lambda_{\text{perc}}\mathcal{L}_{\text{perc}} \\ +\lambda_{\text{style}}\mathcal{L}_{\text{style}},\end{aligned} \tag{12}$$

where the loss weights are set to: $\lambda_{\text{adv}} = 0.1$, $\lambda_{\ell_1} = 1.0$, $\lambda_{\text{perc}} = 0.1$, $\lambda_{\text{style}} = 250$.

4 Experiments

4.1 Experimental Setup

We evaluate on Places2 (12,000 test images, (Zhou et al., 2017)) and CelebA-HQ (2,000 test images, (Karras et al., 2018)), both at 512×512 resolution with irregular masks grouped into six mask-area ranges (0 to 10% through 50 to 60%). We compare SpikeInpaint with other representative image inpainting methods: the FFC-based LaMa (Suvorov et al., 2022), the two-stage methods ZITS (Dong et al., 2022) and CTSDG (Guo et al., 2021), the attention-based models AOT-GAN (Zeng et al., 2022) and Co-ModGAN (Zhao et al., 2021), and the transformer-based MAT (Li et al., 2022). Performance is evaluated with PSNR, SSIM, FID and LPIPS computed on the composited output, together with inference latency measured on a single RTX 4070.

The generator and discriminator are trained with the AdamW optimizer (Loshchilov & Hutter, 2019) ($\beta_1 = 0.0$, $\beta_2 = 0.9$). The generator learning rate is 10^{-3} and the discriminator learning rate is 10^{-4}. The discriminator is a five-layer PatchGAN (Isola et al., 2017) with spectral normalization (Miyato et al., 2018). All models are implemented in PyTorch (Paszke et al., 2019), and spiking neurons are based on SpikingJelly (Fang et al., 2023). All spiking neurons operate with a single timestep. Each forward pass consists of one charge-fire-reset cycle. Membrane states are cleared after every backward pass to avoid state accumulation across mini-batches. Batch normalization is applied before TriHardLIF activation. During inference, a single forward pass is followed by a membrane reset. TriHardLIF uses a threshold of $\theta = 2.0$ in the two SNN Bottleneck modules and $\theta = 1.5$ in the four FFC ResBlock-SNN modules and in SNNGate Light; the corresponding membrane time constants are $\tau = 4.0$, 6.0 and 3.0.

Method	Places2					CelebA-HQ			
	PSNR↑	SSIM↑	FID↓	LPIPS↓	Latency(ms)	PSNR↑	SSIM↑	FID↓	LPIPS↓
LaMa	25.50	0.8600	2.78	0.1114	57.09	27.74	0.8807	8.26	0.1117
Co-ModGAN	23.79	0.8203	3.09	0.1391	10.72	27.87	0.8958	3.78	0.0940
AOT-GAN	25.21	0.8508	5.06	0.1249	14.46	29.04	0.9131	4.17	0.0874
ZITS	25.48	0.8594	2.82	0.1110	376.30	28.50	0.8976	7.07	0.0970
CTSDG	24.88	0.8381	11.63	0.1645	157.50	27.43	0.8743	11.95	0.1471
MAT	24.48	0.8405	2.36	0.1180	76.30	28.54	0.9071	3.31	0.0882
Ours	**24.41**	**0.8222**	**12.41**	**0.1802**	**27.96**	**25.78**	**0.8377**	**24.13**	**0.1790**

Table 1: Quantitative comparison with representative image inpainting methods on Places2 and CelebA-HQ. Quality metrics for the published methods are as reported by their authors under the same irregular-mask protocol and image size 512×512; latency is measured by us on a single RTX 4070, and SpikeInpaint is evaluated by us throughout.

Mask Area	Places2				CelebA-HQ			
	PSNR↑	SSIM↑	FID↓	LPIPS↓	PSNR↑	SSIM↑	FID↓	LPIPS↓
0-10%	32.07	0.9714	4.06	0.0424	34.14	0.9728	5.30	0.0478
10-20%	27.78	0.9355	10.27	0.0829	29.67	0.9410	12.42	0.0876
20-30%	24.78	0.8831	18.78	0.1352	26.60	0.8963	22.48	0.1356
30-40%	22.48	0.8158	29.76	0.1946	24.01	0.8375	35.05	0.1915
40-50%	20.62	0.7351	45.23	0.2608	21.79	0.7625	63.03	0.2551
50-60%	17.91	0.5852	75.53	0.3678	17.84	0.6054	133.69	0.3648
all	24.41	0.8222	12.41	0.1802	25.78	0.8377	24.13	0.1790

Table 2: SpikeInpaint performance by mask ratio on Places2 and CelebA-HQ.

4.2 Quantitative Results

Table 1 reports results on Places2 and CelebA-HQ. On Places2, SpikeInpaint reaches 24.41 dB PSNR in 27.96 ms, against 25.50 dB in 57.09 ms for LaMa, which is a 2.0× latency reduction for 1.09 dB PSNR and a larger advantage over the multi-stage methods (5.6× CTSDG, 13.4× ZITS) and over MAT, which takes 76.30 ms. Co-ModGAN (10.72 ms) and AOT-GAN (14.46 ms) are faster, so we claim no latency advantage across the board. On quality, MAT leads both benchmarks (24.48 dB and 28.54 dB PSNR, 2.36 and 3.31 FID), and SpikeInpaint trails it on every metric; the efficiency claim of this work is spike sparsity rather than latency ranking, since 27.96 ms reflects the dense encoder and decoder, which still execute 80.44% of operations as ANN MACs.

Pixel-level accuracy and distribution-level realism diverge on this task. SpikeInpaint trails the published baselines on FID and LPIPS on both datasets (12.41 vs 2.36 on Places2 and 24.13 vs 3.31 on CelebA-HQ against MAT), which is the metric signature of the piecewise-constant shading that the ternary dead zone produces on smooth regions (Figure 7 in the Appendix): the structure is preserved while the local texture statistics are not. We therefore treat FID and LPIPS as the cost of ternarization rather than as a result of this work. Table 2 breaks this down by mask ratio. PSNR falls from 32.07 dB at 0-10% masks to 17.91 dB at 50-60% on Places2, and from 34.14 to 17.84 dB on CelebA-HQ, with FID rising from 4.06 to 75.53 and from 5.30 to 133.69; the degradation accelerates beyond 30% holes.

Across the four FFC ResBlock-SNN modules the local branch fires at 0.74–3.25% and the global branch at 0.86–4.66%, while the transition modules reach 2.41–10.95%; overall 4.11% of spiking neurons fire, i.e. 95.89% sparsity (Figure 4; Figure 7 in the Appendix).

4.3 Qualitative Results

Qualitative results are presented in Figure 6 of the Appendix, with the top two rows showing Places2 results and the bottom two rows showing CelebA-HQ results. On Places2, SpikeInpaint reconstructs repetitive structures and large missing regions with coherent global layouts, although fine textures are slightly less sharp than those produced by LaMa. Under the largest masks (50-60%) the reconstructed scenes remain globally consistent, but the degradation is no longer confined to texture: Table 2 reports PSNR of 17.91/17.84 dB, SSIM of 0.5852/0.6054 and FID of 75.53/133.69 on the

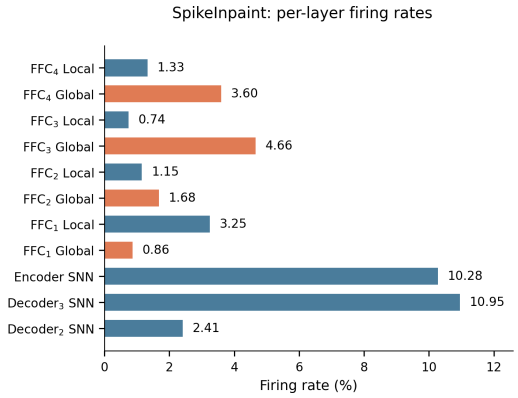


Figure 4: Per-layer firing rates. Blue: local; orange: global; gray: encoder/decoder. Overall firing rate 4.11% (95.89% sparsity).

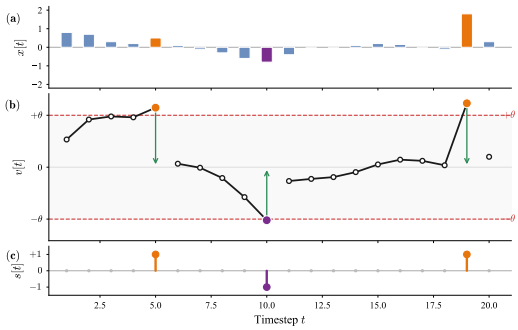


Figure 5: TriHardLIF dynamics. (a) Input current. (b) Membrane potential with threshold (red dashed) and hard reset (green arrows). Shaded region: silent zone ($|v| < \theta$). (c) Spike output. At $T = 1$ inference, each neuron executes a single charge–fire–reset per forward pass.

two datasets, and the faceted shading discussed below becomes the dominant error. On CelebA-HQ, SpikeInpaint preserves facial geometry and semantic consistency, while fine-grained regions such as skins and iris boundaries appear less detailed, reflecting the increased difficulty of recovering high-frequency visual content under sparse spiking representations.

4.4 Energy Analysis

We estimate operation-level energy with the MAC/AC model (Horowitz, 2014) (4.6 pJ per 32-bit MAC, 0.9 pJ per AC) in Table 3. Of the 3.65 T operations, $m = 2.94$ T are dense MACs and $s = 0.71$ T lie in the spiking modules. Charging the spiking operations as ACs discounted by the firing rate ($\rho = 4.11\%$) saves 19.3% over a dense execution, or 15.7% without spike skipping: the saving comes from MAC-to-AC substitution and spike skipping, not from sparsity alone. This is an operation count rather than a timing measurement, and an upper bound, since the FFC global branch (FFT, complex spectral convolution, inverse FFT) remains floating-point regardless of input sparsity.

4.5 Ablation Studies

Architecture evolution. Table 4 reports the architectural variants. Starting from a ResNet-based SNN generator, replacing standard residual blocks with FFC blocks improves PSNR from 24.03 to 24.21 dB and FID from 15.73 to 11.90, which supports separating local spatial features from global Fourier representations. Replacing TriHardLIF with Signed IF lowers PSNR to 24.05 dB, indicating that the ternary dead zone is better suited to stable single-timestep inference. Adding cross-attention

Quantity	Value
Dense ANN MACs	2.94 T (80.44%)
Spiking-module operations	0.71 T (19.56%)
Total operations	3.65 T
Spike sparsity	95.89%
Firing rate	4.11%
Avg. spikes per image	3,203,159
Est. energy per image, all-ANN	16.79 J
Est. energy per image, hybrid	13.55 J
Est. energy reduction	19.3% (upper bound)

Table 3: Operation counts and energy estimate, profiled on a 2,000-image subset of the Places2 test split. The reduction is an upper bound; see Section 4.4.

Structure	PSNR	SSIM	FID	LPIPS
ResNet + SNN	24.03	0.8137	15.73	0.1887
+ FFC	24.21	0.8190	**11.90**	0.1845
+ Signed IF	24.05	0.8121	13.70	0.1829
+ Cross-Attn	23.95	0.8095	13.82	0.1864
w/o SNNGate	23.97	0.8142	13.47	0.1822
Full (Ours)	**24.41**	**0.8222**	12.41	**0.1802**

Table 4: Ablation of generator architecture components on Places2.

lowers PSNR further to 23.95 dB, so additional feature interaction is unnecessary once the FFC backbone covers long-range dependencies; removing SNNGate Light gives 23.97 dB, showing that channel-wise fusion compensates for the information lost in the sparse spiking branch. The full model reaches 24.41 dB, the highest PSNR in the table, although its FID (12.41) is slightly above that of the + FFC variant (11.90): ternarization improves pixel-level accuracy more than texture statistics.

Neuron selection. Table 5 evaluates four neuron formulations under an identical architecture, isolating the contribution of TriHardLIF. The all-ANN model provides an upper bound of 24.50 dB PSNR, and TriHardLIF attains the highest PSNR among the spiking variants (24.41 dB), leaving a 0.09 dB gap to it. Unlike Bipolar Hard LIF, whose responses are always ± 1, TriHardLIF also emits zero, so a unit can be silenced while the branch still carries signals of both signs. Bipolar Hard LIF reaches a lower FID (11.20 vs. 12.41) and a lower LPIPS (0.1753 vs. 0.1802) but 0.31 dB lower PSNR, which we read as a trade-off between perceptual realism and pixel-level fidelity; among the variants compared here, TriHardLIF gives the best pixel-level accuracy.

Neuron	PSNR	SSIM	FID	LPIPS
All-ANN	24.50	0.8261	10.26	0.1718
IF	24.37	0.8171	12.99	0.1836
Bipolar Hard LIF	24.10	0.8160	**11.20**	**0.1753**
TriHardLIF	**24.41**	**0.8222**	12.41	0.1802

Table 5: Comparison of neuron formulations under identical architecture on Places2.

5 Conclusion and Limitations

We introduce SpikeInpaint, a hybrid ANN–SNN generator that confines spiking computation to the Fourier bottleneck and the channel-transition modules on either side of it. Our analysis shows that doing so costs 0.09 dB PSNR and 0.0039 SSIM relative to an otherwise identical all-ANN generator on Places2, and locates the sparsity in the Fourier-domain branches and in the decoder gates. These results indicate that sparse spiking computation can carry dense reconstruction. Two limitations bound the claim. The reported sparsity does not translate into wall-clock savings on standard GPUs, and against published dense inpainting models SpikeInpaint remains behind on FID and LPIPS as well as on the face benchmark (Table 1); the claim supported here is the cost of sparsification relative to the matched dense baseline, not parity with dense inpainting state of the art.

AI USE STATEMENT

Our use of generative AI tools is limited to writing assistance: the authors used large language models to draft and paraphrase passages of the manuscript and to edit text for readability, which fall under the recommended-disclosure categories of the ICLR policy. Generative AI tools were not used to generate synthetic data, to formulate mathematical claims, to assist with proofs, or to design the methodology or experiments. All AI-assisted text was reviewed, verified, and rewritten by the authors, who take responsibility for the final content, claims, and artifacts of this work.

ETHICS STATEMENT

This work develops a generative image inpainting method trained on the public Places2 and CelebA-HQ datasets. Image inpainting can be misused to alter or fabricate visual content. We do not introduce capabilities beyond those of existing inpainting methods, and we note this general risk so that downstream use of such models is informed.

REPRODUCIBILITY STATEMENT

Section 3 gives the generator, discriminator, and spiking-neuron configuration; Section 3.6 gives all losses and weights; Section 4.1 gives datasets, resolution, mask ranges, optimizer, learning rates, and the SpikingJelly single-timestep protocol. The energy, spike-rate, and gate statistics are computed on frozen weights over 2,000 held-out Places2 images, using the MAC/AC constants in Section 4.4. All experiments use a single RTX 4070.

REFERENCES

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

Lu Chi, Borui Jiang, and Yadong Mu. Fast fourier convolution. *Advances in Neural Information Processing Systems*, 33:4479–4488, 2020.

Tianyi Chu, Jiafu Chen, Jiakai Sun, Shuobin Lian, Zhizhong Wang, Zhiwen Zuo, Lei Zhao, Wei Xing, and Dongming Lu. Rethinking fast fourier convolution in image inpainting. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23138–23148. IEEE, 2023.

Qiaole Dong, Chenjie Cao, and Yanwei Fu. Incremental transformer structure enhanced image inpainting with masking positional encoding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11358–11368, 2022.

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.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2414–2423, 2016.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Xiefan Guo, Hongyu Yang, and Di Huang. Image inpainting via conditional texture and structure dual generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 14134–14143, 2021.

Mark Horowitz. 1.1 computing's energy problem (and what we can do about it). In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 10–14. IEEE, 2014. doi: 10.1109/ISSCC.2014.6757323.

Yangfan Hu, Huajin Tang, and Gang Pan. Spiking deep residual networks. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8):5200–5205, 2021.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pp. 694–711. Springer, 2016.

Hiromichi Kamata, Yusuke Mukuta, and Tatsuya Harada. Fully spiking variational autoencoder. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 7059–7067, 2022.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

Seijoon Kim, Seongsik Park, Byunggook Na, and Sungroh Yoon. Spiking-yolo: spiking neural network for energy-efficient object detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 11270–11277, 2020.

Sora Kim, Sungho Suh, and Minsik Lee. Rad: Region-aware diffusion models for image inpainting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 2439–2448, 2025.

Vineet Kumar, A. K. Sinha, and A. K. Solanki. Image inpainting through textures synthesis using spiking neural networks. *International Journal on Emerging Technologies*, 10(4):43–49, 2019.

Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. Mat: Mask-aware transformer for large hole image inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10758–10768, 2022.

Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 85–100, 2018.

Ruijun Liu, Xiangshang Wang, Huimin Lu, Zhaohui Wu, Qian Fan, Shanxi Li, and Xin Jin. Sccgan: style and characters inpainting based on cgan. *Mobile networks and applications*, 26(1):3–12, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11461–11471, 2022.

Xinhao Luo, Man Yao, Yuhong Chou, Bo Xu, and Guoqi Li. Integer-valued training and spike-driven inference spiking neural network for high-performance and energy-efficient object detection. In *European Conference on Computer Vision*, pp. 253–272, 2024.

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

Wei Miao, Jiangrong Shen, Hongming Xu, Tommi Kärkkäinen, Qi Xu, Yi Xu, and Fengyu Cong. Advanced spikingyolox: Extending spiking neural network on object detection with spike-based partial self-attention and 2d-spiking transformer. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 10837–10846, 2025a.

Wei Miao, Jiangrong Shen, Qi Xu, Timo Hamalainen, Yi Xu, and Fengyu Cong. Spikingyolox: Improved yolox object detection with fast fourier convolution and spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 1465–1473, 2025b.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. Edgeconnect: Structure guided image inpainting using edge prediction. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 3265–3274, 2019. doi: 10.1109/ICCVW.2019.00408.

Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2536–2544, 2016.

Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784):607–617, 2019.

Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2149–2159, 2022.

Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience*, 12:331, 2018.

Shiyuan Yang, Xiaodong Chen, and Jing Liao. Uni-paint: A unified framework for multimodal image inpainting with pretrained diffusion model. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 3190–3199, 2023.

Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5505–5514, 2018.

Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4471–4480, 2019.

Yanhong Zeng, Jianlong Fu, Hongyang Chao, and Baining Guo. Aggregated contextual transformations for high-resolution image inpainting. *IEEE transactions on visualization and computer graphics*, 29(7):3266–3280, 2022.

Guanhua Zhang, Jiabao Ji, Yang Zhang, Mo Yu, Tommi Jaakkola, and Shiyu Chang. Towards coherent image inpainting using denoising diffusion implicit models. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 41164–41193, 2023.

Shengyu Zhao, Jonathan Cui, Yilun Sheng, Yue Dong, Xiao Liang, Eric I-Chao Chang, and Yan Xu. Large scale image completion via co-modulated generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2021. Spotlight.

Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li. Going deeper with directly-trained larger spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11062–11070, 2021.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

Shipeng Zhu, Pengfei Fang, Chenjie Zhu, Zuoyan Zhao, Qiang Xu, and Hui Xue. Text image inpainting via global structure-guided diffusion models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 7775–7783, 2024.

A APPENDIX

This appendix provides the qualitative comparisons referenced in Section 4.3 and the per-branch spike statistics behind the firing rates reported in Section 4.2.

Figure 6 shows four masked test images, two from Places2 (rows 1-2) and two from CelebA-HQ (rows 3-4). Columns are ordered as Input, AOT-GAN, Co-ModGAN, CTSDG, LaMa, ZITS, MAT, Ours. Two behaviours are consistent across the examples.

Scene structure is recovered at the level of the baselines. In both Places2 rows, every method reconstructs the global layout of the masked region, and the visible differences between columns are confined to foliage texture. The CelebA-HQ rows show the same at the level of geometry: pose, identity, and the direction of illumination are preserved across the method columns (b)–(h), including MAT in (g) and ours in (h).

The characteristic error of SpikeInpaint is piecewise-constant shading on smooth regions. In row 4 the Ours column renders the forehead and cheek as faceted patches separated by visible boundaries, whereas LaMa and ZITS render continuous skin gradients; row 3 shows a milder version of the same effect. This is the expected visual signature of the ternary dead zone in Eq. 2: features with $|v| < \theta$ are mapped to zero rather than attenuated, so a low-contrast region is reconstructed from a small number of discrete levels. The effect is weak in row 1, where the masked region contains strong structure, and strongest in row 4, where it spans a large smooth area, which matches the per-mask degradation in Table 2 (CelebA-HQ PSNR falls from 29.67 dB at a 10-20% mask to 21.79 dB at 40-50%). We include these examples because they locate the remaining error of the sparse bottleneck, rather than quantifying it.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 6: Qualitative comparison. Left to right: Input, AOT-GAN, Co-ModGAN, CTSDG, LaMa, ZITS, MAT, Ours. Rows 1 and 2: Places2; rows 3 and 4: CelebA-HQ.

Figure 7 localises the firing reported in Section 4.2. Each panel shows the mean spike count per spatial position for one FFC ResBlock-SNN, aggregated over the channels of that branch and over 2,000 held-out test images; the percentage above each panel is the firing rate of that branch, i.e. the fraction of its spiking neurons that emit a non-zero spike.

The two branches differ in how their activity is distributed. Local-branch activity is spatially structured and concentrated: the bright contours in FFC_1 Local follow object boundaries and high-frequency content, and the activity is carried by a small number of locations. Global-branch activity is spatially uniform: the global branch mixes information across the whole image before the activation is applied, so the resulting map is smooth and its threshold crossings are spread over the map

rather than aligned with edges. The branch-level rates (0.74-3.25% local, 0.86-4.66% global) are the aggregate counterpart of this difference, and they are comparable in magnitude: the two branches reach similar firing rates through different spatial organizations.

The dominant branch reverses with depth. FFC_1 fires mainly on the local branch (3.25% vs. 0.86%), whereas FFC_2-FFC_4 fires mainly on the global branch (1.15% vs. 1.68%, 0.74% vs. 4.66%, and 1.33% vs. 3.60%). The crossover between FFC_1 and FFC_2 occurs because the shallow layer contains more local details than the deep layers, which is why we report per-block rates rather than a single local/global average.

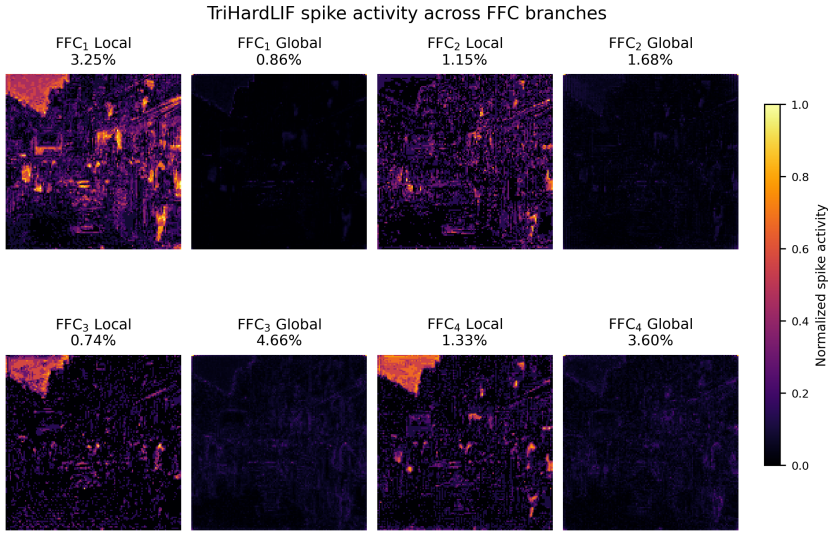


Figure 7: Spike activity across the four FFC ResBlock-SNN modules. Each panel shows the mean spike count per spatial position for one FFC block, normalized independently within that panel; the percentage is the branch firing rate (fraction of spiking neurons emitting a non-zero spike) over 2,000 held-out test images. Local-branch activity follows object boundaries, whereas global-branch activity is spatially diffuse; the dominant branch switches from local to global between FFC_1 and FFC_3.