SpikePack: Enhanced Information Flow in Spiking Neural Networks with High Hardware Compatibility

Supplementary Material

A. Mutual Information Analysis of *SpikePack*

In this appendix, we provide a formal analysis of the mutual information properties of the proposed *SpikePack* neuron model, comparing it with the traditional Leaky Integrate-and-Fire (LIF) neuron model. This analysis aims to show that *SpikePack* neurons retain more information between pre-synaptic inputs and post-synaptic outputs, thereby reducing information loss during spike transmission.

A.1. Problem Statement

Consider a spiking neuron receiving binary input spikes over T time steps from N pre-synaptic neurons. Let $\mathbf{S}^l \in \{0,1\}^{N\times T}$ denote the input spike matrix, where each element $s^l_{n,t}$ represents the spike from the n-th neuron at time step t. Each spike $s^l_{n,t}$ is assumed to be an independent Bernoulli random variable with parameter p, i.e., $s^l_{n,t} \sim \text{Bernoulli}(p)$. The synaptic weights are represented by $\mathbf{w} \in \mathbb{R}^N$, where each weight w_n is drawn independently from a Gaussian distribution $\mathcal{N}(0,\sigma^2)$.

Our objective is to compute and compare the mutual information $I(\mathbf{S}^l;\mathbf{s}^l)$ between input and output spikes for both *SpikePack* and LIF neurons.

A.2. Mutual Information in *SpikePack* Neurons

Accumulated Membrane Potential In the *SpikePack* neuron, the accumulated membrane potential v^l_g is defined as:

$$v^l_g = \mathbf{w}^\top \mathbf{S}^{l-1}\mathbf{q}, \tag{11}$$

where $\mathbf{q} = [\tau^{T-1}, \tau^{T-2}, \ldots, \tau^0]^\top$ incorporates the effect of leakage across time steps.

Distribution of v^l_g Given that the input spikes are independent Bernoulli random variables and the weights are independent Gaussian random variables, the accumulated membrane potential v^l_g is a sum of independent random variables. By the Central Limit Theorem, v^l_g approximates a Gaussian distribution when N is large.

Mean of v^l_g:

$$\mu_{v^l_g} = \mathbb{E}[v^l_g] = \sum_{n=1}^{N}\mathbb{E}[w_n]\sum_{t=1}^{T}\mathbb{E}[s^l_{n,t}]q_t = 0, \tag{12}$$

since $\mathbb{E}[w_n] = 0$.

Variance of v^l_g:

$$\sigma^2_{v^l_g} = \mathbb{E}[{v^l_g}^2] = \sigma^2 Np(1-p)\left(\sum_{t=1}^{T} q_t\right)^2, \tag{13}$$

where $q_t = \tau^{t-1}$.

Differential Entropy of v^l_g Since v^l_g is approximately Gaussian with variance $\sigma^2_{v^l_g}$, its differential entropy is:

$$h(v^l_g) = \frac{1}{2}\log_2(2\pi e\sigma^2_{v^l_g}). \tag{14}$$

Conditional Entropy $h(v^l_g|\mathbf{s}^l)$ The *SpikePack* neuron generates output spikes $\mathbf{s}^l$ by quantizing the continuous membrane potential v^l_g with a quantization step size θ. This process introduces quantization noise, as v^l_g is mapped to the nearest discrete level defined by θ. Following the approach in [45], we assume that this quantization noise is uniformly distributed over $\left[-\frac{\theta}{2},\frac{\theta}{2}\right]$. This assumption is valid when the quantization step size θ is relatively small compared to the variance of v^l_g, and the signal v^l_g is approximately Gaussian and sufficiently random.

Given that the quantization noise q is uniformly distributed over $\left[-\frac{\theta}{2},\frac{\theta}{2}\right]$, the probability density function of q is:

$$f(q) = \begin{cases} \frac{1}{\theta} & \text{for } -\frac{\theta}{2} \le q \le \frac{\theta}{2}, \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

The conditional entropy $h(v^l_g|\mathbf{s}^l)$ represents the uncertainty introduced by quantizing v^l_g and is equal to the entropy of the quantization noise q over the interval $\left[-\frac{\theta}{2},\frac{\theta}{2}\right]$. The entropy of a continuous uniform distribution is calculated as:

$$h(v^l_g|\mathbf{s}^l) = \int_{-\theta/2}^{\theta/2} -f(q)\log_2(f(q))\,dq. \tag{16}$$

Substituting $f(q) = \frac{1}{\theta}$, we get:

$$h(v^l_g|\mathbf{s}^l) = \log_2(\theta). \tag{17}$$

To refine this result, we apply a correction factor for the entropy of the uniform distribution, considering its variance. For a uniform distribution over $\left[-\frac{\theta}{2},\frac{\theta}{2}\right]$, the variance is $\text{Var}(q) = \frac{\theta^2}{12}$ [45]., so the standard deviation is

$\frac{\theta}{\sqrt{12}}$. Thus, the correction term $\log_2(\sqrt{12})$ accounts for the spread of the distribution:

$$h(v_g^l|\mathbf{s}^l) = \log_2(\theta) - \log_2(\sqrt{12}). \tag{18}$$

This refined expression for the conditional entropy $h(v_g^l|\mathbf{s}^l)$ accurately reflects the quantization effects within the *SpikePack* neuron model.

Mutual Information Calculation The mutual information between v_g^l and $\mathbf{s}^l$ is:

$$I(v_g^l; \mathbf{s}^l) = h(v_g^l) - h(v_g^l|\mathbf{s}^l) = \frac{1}{2}\log_2\left(\frac{12\sigma_{v_g^l}^2}{\theta^2}\right). \tag{19}$$

Since $\mathbf{s}^l$ is a deterministic function of v_g^l, we have:

$$I(\mathbf{S}^l; \mathbf{s}^l) = I(v_g^l; \mathbf{s}^l). \tag{20}$$

Thus, the mutual information for the *SpikePack* neuron is:

$$I_{\text{SP}} = \frac{1}{2}\log_2\left(\frac{12\sigma_{v_g^l}^2}{\theta^2}\right). \tag{21}$$

A.3. Mutual Information in LIF Neurons

Approximated Membrane Potential In the LIF neuron, the recursive membrane potential update complicates a direct calculation of mutual information. We approximate the membrane potential at time t as:

$$v_t' = \mathbf{w}^\top \mathbf{s}_t^l, \tag{22}$$

ignoring temporal dependencies and leakage.

Distribution of v_t' Each v_t' is approximately Gaussian with mean zero and variance:

$$\sigma_{v_t'}^2 = \sigma^2 N p(1-p). \tag{23}$$

Probability of Spiking and Entropy of Output Spikes The probability of an output spike at time t is:

$$P(s'_{\text{out},t} = 1) = Q\left(\frac{\theta}{\sigma_{v_t'}}\right), \tag{24}$$

where $Q(\cdot)$ is the Q-function. Using this probability, the entropy of the output spike at each time step is:

$$\begin{aligned} H(s'_{\text{out},t}) = &- P(s'_{\text{out},t} = 1)\log_2 P(s'_{\text{out},t} = 1) \\ &- P(s'_{\text{out},t} = 0)\log_2 P(s'_{\text{out},t} = 0). \end{aligned} \tag{25}$$

Upper Bound on Mutual Information Assuming independence across time steps, the total mutual information is bounded by:

$$I_{\text{LIF}} = I(\mathbf{S}^l; \mathbf{s}^l) \leq \sum_{t=1}^{T} H(s'_{\text{out},t}). \tag{26}$$

A.4. Comparative Analysis and Numerical Estimation

Parameter Settings We use the following parameters for both theoretical and numerical estimation:

- Number of pre-synaptic neurons: $N = 16$
- Number of time steps: $T = 16$
- Weight variance: $\sigma^2 = 1$
- Input spike probability: $p = 0.5$
- Membrane time constant: $\tau = 2$

***SpikePack* Mutual Information Calculation** Compute $\sigma_{v_g^l}^2$ using Eq. (13):

$$\sigma_{v_g^l}^2 = 4\left(2^{16} - 1\right)^2. \tag{27}$$

Substitute $\sigma_{v_g^l}^2$ and $\theta = \frac{6\sigma_{v_g^l}}{2^T}$ into Eq. (21):

$$I_{\text{SP}} \approx 15.21 \text{ bits}. \tag{28}$$

LIF Neuron Mutual Information Calculation For the LIF neuron, $\sigma_{v_t'}^2 = 4$ and $P(s'_{\text{out},t} = 1) = Q(0.5) \approx 0.3085$. Using Eq. (25), each time step contributes approximately $H(s'_{\text{out},t}) \approx 0.881$ bits, leading to:

$$I_{\text{LIF}} \leq 16 \times 0.881 = 14.096 \text{ bits}. \tag{29}$$

Comparison and Interpretation The mutual information estimates indicate that:

- *SpikePack* achieves $I_{\text{SP}} \approx 15.21$ bits.
- LIF Neuron achieves $I_{\text{LIF}} \leq 14.096$ bits.

This demonstrates that *SpikePack* retains more information, validating the theoretical analysis.

A.5. Empirical Validation

To validate our theoretical findings, we conducted Monte Carlo simulations to estimate $I(\mathbf{S}^l; \mathbf{s}^l)$ for both neuron models under various configurations of N and T. The results, depicted in Figure 4, Section 3.2, confirm that *SpikePack* neurons consistently achieve higher mutual information than LIF neurons across different settings, reinforcing the conclusion that *SpikePack* effectively reduces information loss during spike transmission.

This analysis shows that the *SpikePack* neuron model achieves higher mutual information between input and output spikes than the LIF neuron model. By aggregating information across time steps before spike generation, *SpikePack* reduces information loss and enhances transmission efficiency, supporting more effective information flow in SNNs.

B. Experimental Details

In this section, we provide a comprehensive description of the datasets, model architectures, and hyperparameter settings used in our experiments. This includes details on both static image and neuromorphic datasets, as well as specific training configurations for each task.

B.1. Datasets

We evaluate *SpikePack* on both static and neuromorphic datasets to assess its performance across a range of visual tasks.

Static Datasets

- ImageNet [4]: A large-scale image dataset containing over one million images categorized into 1,000 classes. This dataset provides diverse and complex visual content, which is crucial for evaluating classification performance on high-resolution images. For ImageNet, we resize images to 224×224.
- COCO 2017 [27]: A widely-used benchmark for object detection, containing 118,000 training images and 5,000 validation images with 80 object categories. We use this dataset to test *SpikePack* on object detection tasks.
- ADE20K [57]: A semantic segmentation dataset with over 20,000 training images covering 150 classes. ADE20K provides a challenging setup for testing dense pixel-wise prediction tasks, such as segmentation.

Neuromorphic Datasets

- CIFAR10-DVS [21]: A neuromorphic adaptation of CIFAR-10, generated using a Dynamic Vision Sensor (DVS) to capture asynchronous event streams. The dataset consists of 10 classes, matching the original CIFAR-10 categories, with each sample transformed into a sequence of events.
- DVS-Gesture [1]: A dataset designed for gesture recognition, containing hand gestures captured from different individuals under varying lighting conditions. The dataset offers dynamic and complex temporal patterns that challenge spiking models.
- N-Caltech101 [35]: This dataset is a neuromorphic version of the Caltech101 object classification dataset, generated through a DVS camera that records event-based sequences for 101 object categories.

B.2. Hyperparameters and Configuration

For our experiments, we evaluate *SpikePack* in two settings: direct training and ANN-to-SNN conversion.

In the direct training setup, we adhere to the settings used by Zhou et al. [58] for comparability and consistency. For ImageNet datasets, the input resolution is set to 224×224, unless otherwise noted in the main text. Neuromorphic datasets are resized to 48×48 to streamline computational costs. Batch size is dynamically adjusted according to the specific model architecture, maximizing memory usage without exceeding 40 GB of GPU memory. We employ native Automatic Mixed Precision (AMP) for all training processes to balance computational efficiency and memory usage. The initial learning rate is set to 0.001, and models are trained for 300 epochs unless otherwise specified. The membrane time constant τ is set to 2 by default, and threshold θ is dynamically adjusted as $\theta = T/2^T$, where T is the number of time steps. This approach progressively reduces the threshold over time, creating finer divisions of the input signal, which improves information transmission over longer sequences.

For the ANN-to-SNN conversion experiments, we first calibrate θ by selecting 10% of the training data. This subset is used to set θ in a way that minimizes the risk of overflow during inference. For evaluation, this threshold θ remains fixed to ensure stable performance across the entire test set. During conversion, θ is allocated independently for each channel, enabling fine-grained control over the activation dynamics and improving the robustness of the converted SNN model.

The computation of Synaptic Operations (SOP) follows the same procedure as Zhou et al. [58], where SOP is defined as $\text{SOP} = \text{fr} \times \text{FLOPs} \times T$. Here, fr represents the firing rate, or the proportion of spikes generated over the total possible activations, allowing for a direct comparison of energy efficiency across models with different firing dynamics and time steps.

For object detection and semantic segmentation tasks, we apply the ANN-to-SNN conversion approach, given the high accuracy already achieved through this method. This setup maintains the accuracy benefits of the ANN models while allowing efficient deployment in SNN form, leveraging the sparsity and reduced computational costs enabled by *SpikePack*.

C. Hardware Experiments

To evaluate the performance of *SpikePack* neurons in comparison to traditional Leaky LIF neurons on hardware, we designed a customized digital processor resembling a neuromorphic chip. This processor processes binary spike inputs and synaptic weights, performing event-driven accumulation of synaptic currents. The architecture comprises three primary components: (1) a spike address encoder, which encodes pre-synaptic input spikes to addresses for retrieving the corresponding synaptic weights, (2) an array of processing elements (PEs) with vectorized multiplex-accumulate logic, and (3) parallel neuron node logic responsible for generating output spikes, as depicted in Figure.8. The customized architecture builds upon and extends the

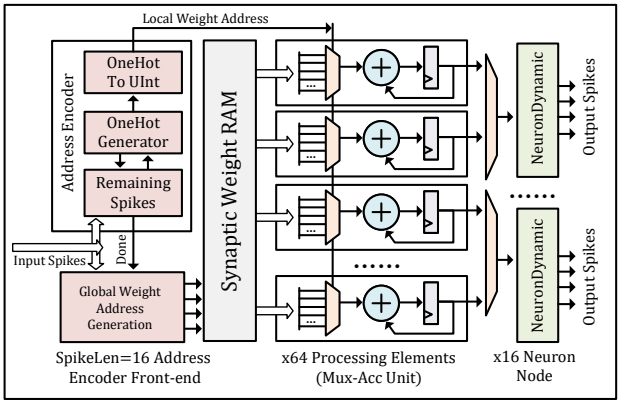

Figure 8. Hardware architecture of the neuromorphic-like processor demo customized for *SpikePack* or LIF neuron.

FireFly-S[24] implementation.

The processor was tailored for both *SpikePack* and LIF neurons, using a shared encoder and PE logic but differing in neuron implementation logic. Table.7 presents the resource consumption of the designs implemented on an XCZU3EG FPGA. In this analysis, we focus on logic resource utilization, excluding on-chip RAM, as synaptic weight data is directly fed from the simulation environment. The device mapping results of two implmentations are shown in Figure.9.

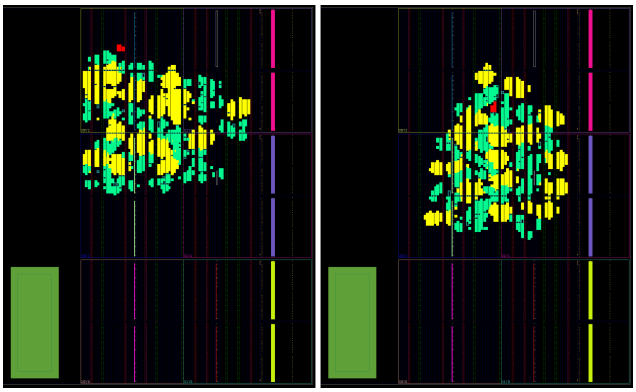

Figure 9. Hardware implementation device map of neuromorphic-like processor for *SpikePack* (right) and LIF (left) neuron on xczu3eg FPGA. Color green area indicates the logic of the processing elements, color yellow indicates the logic of the *SpikePack* or the LIF neuron and color red indicates the logic of the address encoder.

The SpikePack implementation demonstrates a slight reduction in resource consumption compared to the traditional LIF neuron. This efficiency arises from the elimination of the need to store long-term membrane potential in hardware. Additionally, the *SpikePack* implementation consumes less power, operating at 0.808 W compared to 0.816 W for the LIF implementation, both running at 300 MHz.

Table 7. Resource consumption breakdown of customized neuromorphic-like processor for *SpikePack* and LIF neuron.

		LUTs	FFs	CARRY8s
SpikePack	Total	9496	1042	704
	Encode	46	18	0
	PE	4673	1024	256
	Node	4521	0	448
LIF	Total	9850	1302	768
	Encode	46	18	0
	PE	4673	1024	256
	Node	4875	260	512

The ResNet inference latency was measured using a cycle-accurate simulator of the proposed hardware architecture. The spike encoder effectively eliminates redundant spikes, resulting in an inference latency that is strongly correlated with the sparsity level of the spike input. As *SpikePack* inherently produces a more sparse spike output pattern, it achieves lower inference latency and energy per inference compared to the traditional LIF design.