Brain-Inspired Multimodal Spiking Neural Network for Image-Text Retrieval

Supplementary Material

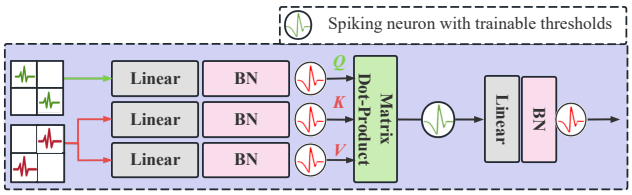

Figure 1. **Detailed structure of our Spike Cross Attention.** Embeddings from both modalities are fused at the spike level via cross-attention matrix multiplication.

A. Spike Fusion Methods

To validate the effectiveness of our Spike Fusion, we design multiple fusion strategies: Spike Cross Attention (SCA), Spike-Concat Self Attention (SCSA), and Spike Comb Cross Attention (SCCA). In this section, we present detailed experiments and analysis of the SCA and SCSA methods.

A.1. Spike Cross Attention

Drawing inspiration from Spikformer [**?**], we propose a purely spiking-driven network called Spike Cross Attention (SCA). Its structure mirrors the conventional cross-attention mechanism, but replaces ReLU activations with LIF neurons and omits the Softmax operation to better align with SNN characteristics.

As shown in Fig. 1, the image spike embedding (green) serves as the *Query*, while the text spike embedding (red) provides both the *Key* and *Value*. All inputs firstly pass through a {`Linear`, `BN`, `LIF`} layer before SCA's matrix operations. During this process, salient elements of the Key activate corresponding *Query* spikes, joint spiking information is retained, and redundant *Query* activations are suppressed. The resulting *Query* spike matrix is thereby fused with cross-modal information while preserving sparsity.

SCA is specifically designed for cross-modal spike sequences: since Q, K, and V are binary spike matrices, their "dot products" reduce to logical AND (&) followed by summation, meeting SNN requirements. Moreover, the order of computation, $(QK^\top)V$ *vs.* $Q(K^\top V)$, can be chosen dynamically to minimize time complexity, selecting between $O(N^2D)$ and $O(ND^2)$. Thus, SCA maintains both biological plausibility and computational efficiency throughout the spike fusion process.

A.2. Spike-Concat Self Attention

To assess the impact of single-stream versus dual-stream architectures on our Spike Fusion, we introduce Spike-Concat Self Attention (SCSA), a single-stream variant. SCSA's structure parallels SCA, with the key difference being input handling: image and text spike embeddings are concatenated along the N and L dimensions before entering the SCSA block:

$$\boldsymbol{X}_{\text{concat}} = \begin{pmatrix} R \\ E \end{pmatrix} = \{r_1; \ldots; r_N; e_1; \ldots; e_L\}, \quad (1)$$

where $\boldsymbol{X}_{\text{concat}} \in \mathbb{R}^{T\times(N+L)\times D}$. Omitting Softmax and scaling factors, SCSA is defined as:

$$\text{SCSA}\left(Q_S, K_S, V_S\right) = Q_S K_S^\top V_S, \quad (2)$$

where Q_S, K_S, and V_S are obtained by linearly projecting X_{concat}:

$$Q_S = X_{\text{concat}} W^Q = \begin{pmatrix} RW^Q \\ EW^Q \end{pmatrix} = \begin{pmatrix} Q_R \\ Q_E \end{pmatrix}, \quad (3)$$

$$K_S = X_{\text{concat}} W^K = \begin{pmatrix} RW^K \\ EW^K \end{pmatrix} = \begin{pmatrix} K_R \\ K_E \end{pmatrix}, \quad (4)$$

$$V_S = X_{\text{concat}} W^V = \begin{pmatrix} RW^V \\ EW^V \end{pmatrix} = \begin{pmatrix} V_R \\ V_E \end{pmatrix}. \quad (5)$$

By matrix multiplication rules:

$$\begin{aligned} Q_S K_S^\top V_S &= \begin{pmatrix} Q_R \\ Q_E \end{pmatrix} \left(K_R^\top K_E^\top\right) \cdot \begin{pmatrix} V_R \\ V_E \end{pmatrix} \\ &= \begin{pmatrix} Q_R K_R^\top Q_R K_E^\top \\ Q_E K_R^\top Q_E K_E^\top \end{pmatrix} \cdot \begin{pmatrix} V_R \\ V_E \end{pmatrix} \\ &= \begin{pmatrix} Q_R K_R^\top V_R + Q_R K_E^\top V_E \\ Q_E K_E^\top V_E + Q_E K_R^\top V_R \end{pmatrix}. \end{aligned} \quad (6)$$

The final SCSA output $\tilde{X} \in \mathbb{R}^{T\times(N+L)\times D}$ is then split according to the original modality dimensions, yielding the fused spike embeddings:

$$R = Q_R K_R^\top V_R + Q_R K_E^\top V_E \in \mathbb{R}^{T\times N\times D}, \quad (7)$$

$$E = Q_E K_E^\top V_E + Q_E K_R^\top V_R \in \mathbb{R}^{T\times L\times D}. \quad (8)$$

This ensures that the outputs R and E both integrate cross-modal information and preserve their own modality-specific features, thereby supporting effective intra- and inter-modal alignment.

A.3. Comparison Results

In Tab. 1, we compare the time and space complexities of our three fusion methods. For SCA and SCSA, the matrix multiplication step incurs $O(ND^2)$ time complexity, and, because both the attention map and the Query matrix must

Table 1. **Time and space complexities of SCA, SCSA, and SCCA.** N (regions) and L (words) are of similar magnitude.

Method	Time Complexity	Space Complexity
SCA	$O(ND^2)$	$O(D^2 + ND)$
SCSA	$O((N+L)D^2)$	$O(D^2 + (N+L)D)$
SCCA	$O(N)$	$O(D)$

Table 2. **Results of different fusion methods on FLICKR30K 1K test set.** R@K denotes Recall@K, and R@Sum is the sum of R@1, R@5, and R@10 for both retrieval directions.

Methods	Image-to-Text			Text-to-Image			R@Sum
	R@1	R@5	R@10	R@1	R@5	R@10	
SCA	81.3	95.8	**98.5**	64.9	88.3	**93.4**	522.1
SCSA	81.0	**96.4**	98.1	64.9	**88.4**	93.2	522.0
SCCA	**82.1**	96.3	98.0	**65.9**	**88.4**	93.2	**523.9**

be stored, $O(D^2 + ND)$ space complexity. By contrast, mask-based Spike Comb Cross Attention traverses only N/h elements per comb and stores just h combs, reducing time complexity to $O(N)$ and space complexity to $O(D)$.

In Tab. 2, we compare the performance of three Spike Fusion methods on FLICKR30K under identical settings. The SCCA-based method achieves the highest accuracy, while the performance gap among the three approaches remains small, demonstrating the effectiveness of spike fusion for cross-modal interaction. Additionally, during training SCCA consumes less memory due to its simpler and more efficient network structure.

B. Pre-trained VLMs

As a general-purpose model, the pretrained Vision-Language Models treat image-text retrieval merely as one of their training objectives to enhance generalization capability. Pre-trained models such as CLIP [**?**], which achieve impressive performance through large-scale data and deep network architectures, are not comparable in retrieval efficiency or energy consumption to specially designed lightweight retrieval models [**?**].

While CLIP employs a ViT backbone to extract image and text features, our method and baselines use Faster R-CNN and BERT for feature extraction, respectively. To further validate the effectiveness of our approach, we also conducted comparison experiments with CLIP. As shown in Tab. 3 , our method (with a single attention block) significantly outperforms CLIP's base variant (12 Transformer blocks) but falls slightly behind the deeper large variant (24 Transformer blocks). This result also indicates that the feature extraction network is interchangeable, and a stronger backbone can lead to improved performance.

Table 3. **Results Compared to Pre-trained Methods.** frcnn means FasterRCNN [**?**].

Methods	Structure	Image-to-Text			Text-to-Image			R@Sum
		R@1	R@5	R@10	R@1	R@5	R@10	
Flickr30K 1K Test Set								
CLIP	ViT-B/32	78.7	95.4	98.0	66.3	88.6	93.1	520.0
CLIP	ViT-L/14	87.3	99.0	99.5	76.4	94.8	97.4	554.5
CMSF	frcnn+bigru	80.7	95.0	97.6	61.3	85.9	91.3	511.8
CMSF	frcnn+bert	82.1	96.3	98.0	65.9	88.4	93.2	523.9
MSCOCO 5K Test Set								
CLIP	ViT-B/32	56.3	81.7	89.4	42.8	71.2	81.1	422.6
CLIP	ViT-L/14	67.1	89.4	94.7	51.6	79.1	87.7	469.6
CMSF	frcnn+bigru	58.5	85.3	92.5	42.5	72.0	82.2	432.0
CMSF	frcnn+bert	61.5	86.7	92.8	45.1	75.0	84.6	445.8

C. Theoretical Energy Consumption

The computational energy consumption on neuromorphic hardware is often measured by operation counts. In ANNs, each operation involves floating-point multiplications and additions (MACs), and the computational cost is estimated by floating-point operations (FLOPs). SNNs, however, are more energy-efficient on neuromorphic hardware since neurons perform only accumulation computations (AC) during spikes, counted as synaptic operations (SyOPs). Following [**?**], the theoretical energy consumption of SNN layer l is:

$$\text{Energy}\,(l) = E_{\text{AC}} \times \text{SOP}_s\,(l)\,. \tag{9}$$

Analogously, for an ANN layer f, the theoretical energy consumption is:

$$\text{Energy}\,(f) = E_{\text{MAC}} \times \text{FLOP}_s\,(f)\,. \tag{10}$$

We assume MAC and AC operations on 45 nm hardware [**?**], with $E_{\text{MAC}} = 4.6$ pJ and $E_{\text{AC}} = 0.9$ pJ (1 J = 10^3 mJ = 10^12 pJ). The number of synaptic operations in SNN layer l is estimated as

$$\text{SOP}_s\,(l) = T \times \text{Rate} \times \text{FLOP}_s\,(l)\,, \tag{11}$$

where T is the number of time steps and Rate is the firing rate of the input spike train at layer l.

In Tab. 4, we present the theoretical energy consumption formulas for each spiking neuron layer in our CMSF network. This includes the Linear layers used to align input dimensions, the Linear projections for Q, K, and V matrices, the matrix multiplications in the three Spike Fusion schemes (SCA, SCSA, SCCA), and the output projection Linear layers. Notably, for the Spike Gated MLP, the gated matrix multiplication acts as a masking operation and therefore incurs negligible energy cost on neuromorphic hardware.

Table 4. **Detailed calculation formulas.** The theoretical energy consumption of each CMSF layer.

Neural Layer	Theoretical Consumption
Region Linear	$E_{\mathrm{AC}} \cdot T \cdot R_r \cdot FL_r$
Word Linear	$E_{\mathrm{AC}} \cdot T \cdot R_w \cdot FL_w$
Q, K, V	$E_{\mathrm{AC}} \cdot T \cdot R_0 \cdot 3ND^2$
SCA	$E_{\mathrm{AC}} \cdot T_1 \cdot R_1 \cdot ND^2$
SCSA	$E_{\mathrm{AC}} \cdot T_2 \cdot R_2 \cdot (N+L)D^2$
SCCA	$E_{\mathrm{AC}} \cdot T_3 \cdot R_3 \cdot N$
Out Linear	$E_{\mathrm{AC}} \cdot T \cdot R_o \cdot FL_o$
Gate Linear	$E_{\mathrm{AC}} \cdot T \cdot R_g \cdot FL_g$
Gate Multiply	0

Table 5. **Ablation of Spike Generator designs on Flickr30K 1K.** R@K denotes Recall@K for image-to-text (first three columns) and text-to-image (next three columns); R@Sum is the total.

Spike Generator	Image-to-Text			Text-to-Image			R@Sum
	R@1	R@5	R@10	R@1	R@5	R@10	
Conv-BN	77.9	95.8	**98.6**	60.8	85.8	91.5	510.4
Delta-BN	76.1	95.4	97.8	58.5	84.0	90.2	502.0
Linear-BN	80.3	**96.5**	98.2	64.8	87.5	92.7	520.0
Linear-LN	81.3	96.1	97.7	64.4	**88.6**	93.2	521.3
Repeat-BN	80.2	95.9	97.5	64.4	88.2	**93.4**	519.6
Repeat-LN	**82.1**	96.3	98.0	**65.9**	88.4	93.2	**523.9**

D. Spike Generator

The Spike Generator, positioned at the front of the network, converts continuous-valued inputs X_f into discrete spike trains X_s and plays a crucial role in preserving semantic information. A simple approach is to repeat the original features T times before neuronal activation. Prior work, such as [**?**], has proposed Delta and Convolution Generators for mapping floating-point time-series data to spike trains; however, these methods may not effectively retain intra-modal semantic structure in multi-modal image-text retrieval tasks. To address this, we explore various combinations of dimensional-expansion and normalization techniques to identify an generator that best preserves semantic information for downstream processing. Our final design is:

$$X_s = \mathcal{SN}\left(\mathrm{LN}\left(\mathrm{Repeat}\left(X_f, T\right)\right)\right), \tag{12}$$

where $\mathrm{Repeat}(X_f, T)$ duplicates features T times, LN denotes layer normalization, and $\mathcal{SN}$ is the LIF neuron. This sequence can be represented as $\{\texttt{Linear-LN-LIF}\}$ and abbreviated as $\{\texttt{Linear-LN}\}$. Ablation studies in Tab. 5 confirm the effectiveness of this Spike Generator design. The experimental results demonstrate that the proposed method achieves significant improvements in both Recall@1 and Recall@Sum metrics.

Table 6. **Ablation of comb head numbers on FLICKR30K 1K.** R@K denotes Recall@K for image-to-text (columns 2-4) and text-to-image (columns 5-7); R@Sum is the total across all six metrics.

Head Num	Image-to-Text			Text-to-Image			R@Sum
	R@1	R@5	R@10	R@1	R@5	R@10	
$h=2$	79.4	**96.5**	97.7	63.9	88.0	93.0	518.6
$h=3$	80.5	96.1	98.0	64.4	**88.5**	**93.3**	520.8
$h=4$	80.2	96.3	**98.2**	64.9	88.4	93.2	521.2
$h=6$	**82.1**	96.3	98.0	**65.9**	88.4	93.2	**523.9**
$h=9$	80.2	95.9	98.0	64.3	88.3	92.9	519.5
$h=12$	80.4	96.5	98.1	64.5	88.3	93.2	520.9
$h=18$	80.0	96.4	97.7	64.1	88.2	93.0	519.3

E. Comb Teeth

Inspired by [**?**], we introduce Spike Comb Cross Attention to enable spike-level fusion. We split the *Query* matrix into h sub-matrices, sum each to obtain h vectors, and then apply spiking neuron activation. These vectors reveal distribution patterns across the embedding dimension D, analogous to the "teeth" of a comb: when they "comb through" the *Key* matrix, they align the spike distributions of *Query* and *Key*, encouraging both modalities to fire at corresponding positions. This alignment enhances cross-modal integration.

To assess the effect of the number of comb "teeth" (h) on modality alignment, we conduct an ablation study (see Tab. 6). Since both the number of regions N and the number of words L are 36 in our experiments, h must be a divisor of 36, *i.e.*, $h \in \{2, 3, 4, 6, 9, 12, 18\}$. We find that a large number of combs (h) causes each comb to cover only a few tokens (regions or words), which is insufficient to capture phrase-level semantics or composite region patterns. Conversely, a small h forces each comb to compress a large amount of fine-grained information, effectively reducing all tokens to a global representation and weakening fine-grained alignment. Therefore, a balance is needed: we observe that $h = 6$ offers the best trade-off and achieves optimal performance.

F. Visualization and Case Study

To further demonstrate CMSF's superiority, we visualize retrieval results on FLICKR30K test sets (see Fig. 2). For each image query, we present the top-10 retrieved sentences; for each text query, we show the top-5 retrieved images. For image queries, CMSF correctly retrieves all relevant sentences. Even when retrieval errors occur, the incorrect sentences come from the same image (highlighted by a red box in the top-left), in both image-text pairs, key words or salient regions such as "men", "group of people", and "standing" appear consistently, indicating that our model returns images with highly similar scene, content, and composition. This highlights CMSF's accuracy and robustness in interpreting

Figure 2. **Visualization of retrieval results.** Top: image-to-text retrieval examples. Bottom: text-to-image retrieval examples by CMSF on FLICKR30K.

image content.

For text queries, CMSF consistently ranks the ground-truth image first. In cases where non-ground-truth images appear, the top results still contain objects matching key terms (*e.g.*, "woman and man", "street", "gray van", "cars"), demonstrating that CMSF reliably captures object-level semantics.