Effet d'un masquage d'impulsions sur la consommation énergétique d'un SNN : application au Spikformer

Oumaima MARSI[1] Sébastien AMBELLOUIS[1] José MENNESSON[2] Cyril MEURIE[1] Anthony FLEURY[2] Charles TATKEU[1]

[1]Univ. Gustave Eiffel, COSYS, LEOST, F-59000 Lille, France

[2]IMT Nord Europe, Institut Mines-Télécom, Univ. Lille, Centre for Digital Systems, F-59000 Lille, France

Résumé – Les réseaux de neurones impulsionnels offrent une alternative économe en énergie aux réseaux neuronaux traditionnels. Cependant, bien que les Transformers impulsionnels, comme Spikformer, soient naturellement plus économes en énergie que leurs homologues non impulsionnels, ils restent coûteux en calcul et redondants dans le traitement des impulsions. Cette étude évalue l'impact de la méthode Random Spike Masking (RSM) qui vise à masquer les impulsions redondantes et ainsi réduire sa consommation énergétique. RSM est appliqué sur Spikformer, une architecture de type Transformer totalement impulsionnelle. L'évaluation porte sur les ensembles de données neuromorphiques (CIFAR10-DVS, DVS Gesture) et montre que RSM réduit la consommation d'énergie jusqu'à 70% avec une perte de précision négligeable.

Abstract – Spiking neural networks offer an energy-efficient alternative to traditional neural networks. However, although Spiking Transformers, such as Spikformer, are naturally more energy-efficient than their non-spiking counterparts, they remain computationally expensive and redundant in processing spikes. This study, evaluates the impact of the Random Spike Masking (RSM) method, which aims at masking redundant spikes and thus at reducing energy consumption. RSM is applied to Spikformer, a fully spiking transformer architecture. The evaluation is done on neuromorphic datasets (CIFAR10-DVS, DVS Gesture) and show that RSM reduces energy consumption by up to 70% with a negligible accuracy loss.

1 Introduction

Les réseaux de neurones impulsionnels (SNNs) sont une alternative écoénergétique aux réseaux neuronaux traditionnels en reproduisant la dynamique des systèmes neuronaux biologiques. Leur capacité à traiter les données de manière asynchrone et événementielle les rend particulièrement adaptés aux applications neuromorphiques et temps-réel [7].

Les Transformers et leur mecanisme d'attention ont démontré des performances exceptionnelles dans le développement des LLM puis en vision par ordinateur [3]. Cependant, l'intégration de l'auto-attention aux SNNs est un défi, car elle repose sur des opérations matricielles incompatibles avec la nature événementielle des SNNs [10]. Plusieurs approches ont été proposées pour adapter ces modèles, notamment à travers le Spiking Self-Attention (SSA) et d'autres variantes hybrides [9].

Le Random Spike Masking (RSM), proposé pour les architectures impulsionnelles dérivées d'un ANN, réduit aléatoirement les impulsions transmises afin d'alléger la charge computationnelle tout en conservant la précision [8].

Dans cet article, nous explorons pour la première fois son application à Spikformer [10], un Transformer entièrement impulsionnel, en analysant ses effets sur la consommation énergétique et la précision, tant en entraînement qu'en inférence, contrairement aux approches ANN-to-SNN classiques [8, 10].

Les principales contributions de cet article peuvent être résumées comme suit :

- Analyse de l'impact du RSM sur Spikformer qui complète l'analyse faite sur un modèle ANN-to-SNN ;
- Validation expérimentale sur des bases de données neuromorphiques (CIFAR10-DVS, DVS Gesture) pour mesurer son effet sur l'efficacité énergétique et les performances du modèle ;
- Réduction de la consommation énergétique jusqu'à 50% avec un masquage de 70%, sans perte significative de précision, confirmant RSM comme une stratégie d'optimisation efficace pour Spikformer.

2 État de l'art

2.1 Réseaux de neurones impulsionnels

Les réseaux de neurones impulsionnels traitent l'information sous forme d'impulsions plutôt que de valeurs continues comme dans les réseaux traditionnels. Grâce à leur dynamique temporelle, où l'encodage repose sur le moment et la fréquence des impulsions, ils offrent une meilleure efficacité énergétique et un traitement asynchrone adapté aux architectures neuromorphiques [7].

Dans un réseau de neurones impulsionnels, chaque neurone accumule un potentiel membranaire U_{mem}en fonction des entrées synaptiques I_{in}. Lorsque le potentiel atteint le seuil U_{th}, une impulsion est émise et le potentiel est réinitialisé. Parmi les modèles proposés dans la littérature, le Leaky Integrate-and-Fire (LIF) [7] est largement adopté pour sa simplicité computationnelle et sa fidélité aux dynamiques biologiques (voir Figure 1). Il reproduit le fonctionnement neuronal en intégrant un mécanisme de fuite, empêchant ainsi une accumulation excessive du potentiel membranaire et la création

d'impulsions inutiles et améliorant l'efficacité énergétique des SNNs.

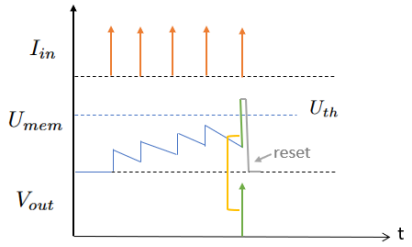

FIGURE 1 : Schéma illustrant le fonctionnement du modèle LIF.

L'apprentissage des réseaux de neurones impulsionnels (SNNs) est complexe en raison de la non-différentiabilité des impulsions, rendant inapplicables les méthodes classiques de descente de gradient. Deux approches permettent de contourner cette contrainte : la conversion ANN-to-SNN qui consiste à transférer les poids d'un ANN pré-entraîné [2], et l'approximation des gradients qui fait appel à une fonction de substitution des gradients différentiable et permet l'entraînement des SNNs via la descente de gradient (Surrogate Gradient Descent - SGD) [6]. Sur le plan de la consommation énergétique, la conversion ANN-vers-SNN génère davantage d'impulsions, ce qui augmente l'énergie consommée. En revanche, l'utilisation de gradients approximés permet d'optimiser efficacement les SNNs par rétropropagation, tout en tirant parti d'une activité neuronale plus éparse. Grâce à l'optimisation par descente de gradient stochastique (SGD), le SNN apprend à ne produire que les impulsions nécessaires à la transmission de l'information, ce qui réduit le calcul de charge et diminue ainsi la consommation énergétique
[6].

2.2 Transformers impulsionnels

Développer des architectures de type Transformers à l'aide de SNNs permet d'optimiser la transmission des impulsions en exploitant leur capacité à modéliser des dépendances à long terme tout en profitant de l'efficacité énergétique des SNNs et de leur faible latence. Contrairement aux Transformers classiques, qui reposent sur une opération de softmax pour calculer l'attention, dans le cas des SNNs une adaptation est requise afin de le faire fonctionner avec des données discrètes. Les premières approches ont cherché à convertir des Transformers ANN en SNNs en conservant le mécanisme d'attention classique fondé sur le softmax [5]. C'est dans [8] que les auteurs ont alors proposé de limiter la création d'impulsions redondantes en utilisant le RSM sur les SNNs créés à partir d'ANNs pré-entrainés. Cependant, ces méthodes requièrent un nombre élevé de pas temporels i.e. des unités de temps discrètes sur lesquelles les impulsions sont intégrées et traitées dans le réseau neuronal [7] augmentant ainsi le nombre d'opérations à effectuer et la consommation énergétique induite.

Plutôt que d'appliquer un softmax sur des valeurs continues, [10] introduit un mécanisme d'attention entièrement impulsionnel appelé Spiking Self-Attention (SSA) remplaçant le softmax par un mécanisme d'accumulation directe des impulsions. Ainsi SSA est compatible avec la dynamique et la nature parcimonieuse des SNNs. Il nécessite moins d'impulsions et requiert un nombre de pas temporels plus faible.

Toutefois, il présente encore une redondance importante dans la transmission des impulsions. Cette redondance provient du fait que, dans un mécanisme d'attention, plusieurs neurones peuvent coder des informations similaires, ce qui entraîne une augmentation du nombre d'impulsions et, par conséquent, une consommation énergétique plus élevée. Dans l'optique de réduire ces redondances, nous proposons d'appliquer la méthode Random Spike Masking (RSM) au Spikformer en exploitant les corrélations spatiales et temporelles des impulsions. RSM est appliqué lors des phases d'entraînement et d'inférence.

3 Méthodologie

3.1 Architecture globale

La figure 2 illustre l'architecture globale de Masked Spikformer, basée sur Spikformer [10] : cette architecture est construite à partir des quatre modules suivants : le Spiking Patch Splitting (SPS) [10], le Spiking Self-Attention (SSAM), le Masked MLP, et la couche de classification linéaire. Le SSAM et le Masked MLP sont le résultat de l'application de RSM aux modules SSA et MLP comme présenté sur la figure 2. Le RSM diffère d'une couche de dropout en ce qu'il est utilisé pendant l'entraînement et pendant l'inférence.

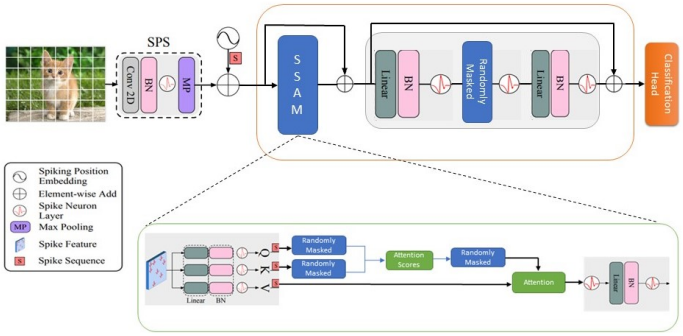

FIGURE 2 : Architecture globale de Masked Spikformer (adapté de [10]).

3.2 RSM appliqué à SSA

Dans Spikformer, le SSA capture les dépendances entre neurones via les requêtes (Q), clés (K) et valeurs (V) encodées sous forme d'impulsions neuronales. Le Random Spike Masking (RSM) applique un masquage binaire aléatoire sur les matrices Q, K, ainsi que sur les scores d'attention issus du produit $QK^\top$, afin de réduire la charge computationnelle et supprimer les interactions redondantes. Le taux de masquage τ contrôle la proportion d'impulsions masquées.

Sur la Figure 2, *Linear* désigne une projection linéaire et *BN* une normalisation par batch. Les couches *Randomly Masked* illustrent l'application du masque binaire. Seules les impulsions non masquées sont transmises à la couche suivante,

réduisant ainsi le nombre de calculs effectués.

Les impulsions retenues sont ensuite pondérées par les scores d'attention masqués, puis passent par une couche linéaire et une normalisation BN, afin d'assurer une représentation adaptée à la classification. *Linear+BN* assure que la sortie de l'attention reste dans un espace de représentation adapté aux couches suivantes du réseau, facilitant ainsi l'apprentissage et la convergence du modèle.

Dans Masked Spikformer, le taux de masquage τ est fixé à l'avance (ex. 20 %, 50 %, 70 %) et appliqué uniformément dans toutes les couches concernées (SSA et MLP), garantissant une réduction contrôlée de l'activité neuronale sans adaptation dynamique.

3.3 Masked MLP

Comme illustré sur la Figure 2, RSM est également appliqué au MLP après la première projection linéaire et la normalisation (BN), avant la propagation vers les couches suivantes. Ce masquage statique, appliqué aussi bien à l'entraînement qu'à l'inférence, induit une forme de sparsité contrôlée : une proportion $1 - \tau$ des activations est désactivée (soit environ 30 %, 50 % ou 80 % selon le taux τ choisi). Certaines connexions ou activations deviennent ainsi systématiquement moins utilisées pendant l'apprentissage. Dans ce contexte, une étape de *pruning* pourrait éliminer définitivement les paramètres les moins informatifs lors de la phase de classification, conduisant à un modèle plus compact et potentiellement plus rapide lors de l'inférence.

4 Expériences

Nous évaluons notre méthode Masked SpikFormer (MS) sur les jeux de données CIFAR10-DVS et DVS Gesture. L'architecture repose sur deux blocs encodeurs, avec un nombre de pas temporels variant entre 8 et 16 afin de maintenir une faible latence. L'entraînement est réalisé sur 96 époques pour CIFAR10-DVS et 192 pour DVS Gesture, à l'aide de l'optimiseur AdamW. Nos implémentations s'appuient sur les frameworks SpikingJelly et PyTorch.

Pour estimer la consommation énergétique, nous suivons la méthode proposée dans [10], en tenant compte des opérations synaptiques effectuées sur une cible CMOS 45 nm. Les coûts énergétiques unitaires sont fixés à $E_{MAC} = 4.6\,\text{pJ}$ pour les multiplications-accumulations, et $E_{AC} = 0.9\,\text{pJ}$ pour les accumulations simples. L'effet du masquage aléatoire est intégré dans le calcul du nombre d'opérations synaptiques par couche l selon :

$$\text{SOPs}(l) = fr \times T \times \text{FLOPs}(l) \times (1 - \tau)$$

où fr est le taux de décharge neuronal, T le nombre de pas temporels, et τ le taux de masquage appliqué dans les modules SSA et MLP. La consommation énergétique totale estimée du modèle est donnée par :

$$E_{\text{MS}} = E_{\text{MAC}} \cdot FL_1 + E_{\text{AC}} \cdot \sum \text{SOP}$$

où FL_1 correspond aux opérations de la première couche convolutionnelle, et $\sum$ SOP regroupe les contributions des autres couches : convolutions, MLP et auto-attention masquée. Cette formulation permet de quantifier l'impact direct du masquage sur la charge computationnelle et la consommation énergétique.

4.1 Jeu de données

CIFAR10-DVS [4] est un jeu de données neuromorphique dérivé d'un ensemble d'images statiques comportant 10 catégories. Les échantillons y sont convertis en événements neuromorphiques à l'aide d'une caméra DVS, générant 9000 échantillons d'apprentissage et 1000 échantillons de test.
DVS128 Gesture [1] est un ensemble de données dédié à la reconnaissance de gestes, comprenant 11 catégories de gestes de la main, effectués par 29 individus sous trois conditions d'éclairage différentes.

4.2 Comparaison de Spikformer avec et sans masquage

Les tableaux 1 et 2 comparent SpikFormer et Masked SpikFormer sur CIFAR10-DVS et DVS Gesture, en termes de précision (Acc), d'énergie (E), et de taille du modèle (Param). ΔE indique la variation énergétique (%) par rapport à SpikFormer. T est le nombre de pas temporels et M le taux de masquage.

Sur CIFAR10-DVS, Masked Spikformer (MS) surpasse systématiquement SpikFormer en précision et en efficacité énergétique. À $T = 16$, un masquage de 20% permet d'atteindre 83.20 % (+2.3 %) avec 28.8 % d'énergie en moins. Avec 70% de masquage, l'économie atteint 72.7 % pour une précision toujours élevée 82.10 %. Réduire le nombre de pas temporels ($T = 10$ ou 8) abaisse encore la consommation (jusqu'à 92.6 % à $T = 8$, $\tau = 70\%$) sans perte significative de performance. Ces gains s'expliquent par la redondance des événements dans CIFAR10-DVS, issus de la conversion d'images statiques, que le RSM parvient à masquer efficacement en introduisant une forme de sparsité utile.

Model	**T**	**M** (%)	**Acc** (%)	**E** (mJ)	**ΔE** (%)	**Param** (M)
SpikFormer	16	-	80.9	8.20	-	2.59
MS (ours)	8	20	81.3	1.62	80.24	2.59
MS (ours)	8	50	80.8	0.98	88.02	2.59
MS (ours)	8	70	80.3	**0.61**	92.56	2.59
MS (ours)	10	20	<u>82.1</u>	2.51	69.39	2.59
MS (ours)	10	50	81.4	1.54	81.19	2.59
MS (ours)	10	70	81.5	0.92	88.78	2.59
MS (ours)	16	20	**83.20**	5.84	28.78	2.59
MS (ours)	16	50	82.70	3.69	55.00	2.59
MS (ours)	16	70	82.10	2.24	72.68	2.59

TABLE 1 : Performances comparées de SpikFormer et Masked SpikFormer (MS) sur CIFAR10-DVS pour différentes valeurs de T et M.

Sur DVS Gesture (Table 2), Masked SpikFormer (MS) offre une excellente robustesse, atteignant jusqu'à 98.61 % de précision avec T=16 et M=70%, surpassant légèrement SpikFormer tout en réduisant la consommation énergétique de 68.9 %. Notons que même avec T=8, la précision reste compétitive ($\geq$95.48 %) et la consommation énergétique diminue

Model	T	M (%)	Acc (%)	E (mJ)	ΔE (%)	Param (M)
SpikFormer	16	-	98.3	9.92	-	2.59
MS (ours)	8	20	95.48	1.714	82.71	2.59
MS (ours)	8	50	95.48	1.031	89.60	2.59
MS (ours)	8	70	96.18	**0.770**	92.24	2.59
MS (ours)	10	20	95.83	2.7695	72.08	2.59
MS (ours)	10	50	96.18	1.8017	81.83	2.59
MS (ours)	10	70	97.22	1.0811	89.10	2.59
MS (ours)	16	20	98.26	7.29	26.52	2.59
MS (ours)	16	50	98.26	4.81	51.41	2.59
MS (ours)	16	70	**98.61**	3.09	68.85	2.59

TABLE 2 : Performances comparées de SpikFormer et Masked SpikFormer (MS) sur DVS Gesture pour différentes valeurs de T et M.

de manière significative jusqu'à 0.770 mJ, soit plus de 92 % d'économie énergétique. Ces résultats illustrent la capacité de MS à conserver de hautes performances sous des contraintes computationnelles sévères.

4.3 Comparaison avec l'état de l'art

Nous comparons désormais notre approche Masked SpikFormer à d'autres méthodes de l'état de l'art, notamment le Masked Spiking Transformer (MST) proposé dans [8], mais uniquement en termes de précision, car aucun calcul énergétique n'a été réalisé pour ces deux jeux de données. Cette architecture ANN-to-SNN atteint des précisions de 86.60% avec 128 pas temporels, 87.20% avec 256, et 88.12% avec 512 pas temporels sur CIFAR10-DVS. En comparaison, notre modèle MS atteint une précision moins élevée de 83.20% mais avec seulement 16 pas temporels donc une latence moins élevée.

Cette différence s'explique en partie par le fait que bien que MST applique RSM pour réduire la redondance importante présente dans les sorties converties du modèle ANN, MST reste dépendant d'un grand nombre de pas temporels pour placer en entrée un nombre suffisant d'impulsions. Par ailleurs, face à ce constat, MST pourrait faire état d'une consommation énergétique plus élevée que notre modèle :notre méthode reposant sur un modèle SNN nativement impulsionnel, plus parcimonieux et nécessitant moins d'impulsions en entrée.

5 Conclusions

Les résultats obtenus avec Masked Spikformer montrent qu'il est possible de réduire significativement la consommation énergétique des réseaux de neurones impulsionnels tout en maintenant une précision compétitive, grâce à l'introduction du Random Spike Masking. Ce gain d'efficacité est d'autant plus marqué lorsque le nombre de pas temporels T est réduit, permettant des représentations temporelles plus compactes. Sur DVS Gesture, le masquage améliore l'efficacité sans compromettre les performances, tandis que sur CIFAR10-DVS, il contribue même à une meilleure précision. Cette approche ouvre ainsi des perspectives prometteuses pour d'autres architectures SNN plus récentes, où son impact sera évalué. Elle pourrait également renforcer la robustesse face au bruit impulsionnel élevé.

Références

[1] Arnon AMIR, Brian TABA, David BERG, Timothy MELANO, Jeffrey MCKINSTRY, Carmelo DI NOLFO, Tapan NAYAK, Alexander ANDREOPOULOS, Guillaume GARREAU, Marcela MENDOZA *et al.* : A low power, fully event-based gesture recognition system. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7243–7252, 2017.

[2] Yongqiang CAO, Yang CHEN et Deepak KHOSLA : Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision*, 113:54–66, 2015.

[3] Kai HAN, Yunhe WANG, Hanting CHEN, Xinghao CHEN, Jianyuan GUO, Zhenhua LIU, Yehui TANG, An XIAO, Chunjing XU, Yixing XU *et al.* : A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):87–110, 2022.

[4] Hongmin LI, Hanchao LIU, Xiangyang JI, Guoqi LI et Luping SHI : Cifar10-dvs : an event-stream dataset for object classification. *Frontiers in neuroscience*, 11:309, 2017.

[5] Etienne MUELLER, Viktor STUDENYAK, Daniel AUGE et Alois KNOLL : Spiking transformer networks : A rate coded approach for processing sequential data. *In 2021 7th International Conference on Systems and Informatics (ICSAI)*, pages 1–5. IEEE, 2021.

[6] Emre O NEFTCI, Hesham MOSTAFA et Friedemann ZENKE : Surrogate gradient learning in spiking neural networks : Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

[7] João D NUNES, Marcelo CARVALHO, Diogo CARNEIRO et Jaime S CARDOSO : Spiking neural networks : A survey. *IEEE access*, 10:60738–60764, 2022.

[8] Ziqing WANG, Yuetong FANG, Jiahang CAO, Qiang ZHANG, Zhongrui WANG et Renjing XU : Masked spiking transformer. *In Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1761–1771, 2023.

[9] Man YAO, Jiakui HU, Zhaokun ZHOU, Li YUAN, Yonghong TIAN, Bo XU et Guoqi LI : Spike-driven transformer. *Advances in neural information processing systems*, 36:64043–64058, 2023.

[10] Zhaokun ZHOU, Yuesheng ZHU, Chao HE, Yaowei WANG, Shuicheng YAN, Yonghong TIAN et Li YUAN : Spikformer : When spiking neural network meets transformer. *In Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023.