SPACE: SPike-Aware Consistency Enhancement for Test-Time Adaptation in Spiking Neural Networks

Anonymous Author(s)
Affiliation
Address
email

A Kernel-Based Consistency Regularization

To investigate whether higher-dimensional feature relationships can enhance the consistency measure, we integrate kernel embeddings into the loss function. The channel-wise feature probability distribution $\mathbf{P}_c(\mathbf{x}_i)$ can be mapped into a higher-dimensional reproducing kernel Hilbert space (RKHS) $\mathcal{H}$, using a Gaussian kernel defined as

$$k(\mathbf{z}, \mathbf{z}') = \exp(-\frac{\|\mathbf{z} - \mathbf{z}'\|^2}{2\sigma^2}), \tag{6}$$

where σ is the kernel bandwidth. The kernel function $k(\mathbf{z}, \mathbf{z}')$ implicitly defines a mapping ϕ: $\mathcal{Z} \to \mathcal{H}$, such that the inner product in $\mathcal{H}$ is given by the kernel:

$$k(\mathbf{z}, \mathbf{z}') = \langle \phi(\mathbf{z}), \phi(\mathbf{z}') \rangle_{\mathcal{H}}. \tag{7}$$

This allows us to compute relationships between feature vectors in a high-dimensional space without explicitly constructing $\phi(\cdot)$. To align feature distributions from augmented samples, we use the Mean Embedding of Distributions, which maps $\mathbf{P}_c(\mathbf{x}_i)$ into a single point $\mu_{\mathbf{P}_c(\mathbf{x}_i)}$ in RKHS. Specifically, the mean embedding is defined as

$$\mu_{\mathbf{P}_c(\mathbf{x}_i)} = \mathbb{E}_{\mathbf{z}_i \sim \mathbf{P}_c(\mathbf{x}_i)}[\phi(\mathbf{z}_i)]. \tag{8}$$

Given samples $\mathbf{z}_i \sim \mathbf{P}_c(\mathbf{x}_i)$, the embedding can be approximated as

$$\mu_{\mathbf{P}_c(\mathbf{x}_i)} \triangleq \frac{1}{D} \sum_{d=1}^{D} \phi(\mathbf{z}_{i_d}). \tag{9}$$

To measure the discrepancy between distributions $\mathbf{P}_c(\mathbf{x}_i)$ and $\mathbf{P}_c(\mathbf{x}_j)$ in RKHS, we employ the Maximum Mean Discrepancy (MMD):

$$\text{MMD}^2(\mathbf{P}_c(\mathbf{x}_i), \mathbf{P}_c(\mathbf{x}_j)) = \|\mu_{\mathbf{P}_c(\mathbf{x}_i)} - \mu_{\mathbf{P}_c(\mathbf{x}_j)}\|^2_{\mathcal{H}}. \tag{10}$$

Using the kernel trick, this can be computed without explicitly constructing $\phi(\cdot)$:

$$\text{MMD}^2(\mathbf{P}_c(\mathbf{x}_i), \mathbf{P}_c(\mathbf{x}_j)) = \frac{1}{D^2} \| \sum_{d=1}^{D} \phi(\mathbf{z}_{i_d}) - \sum_{d=1}^{D} \phi(\mathbf{z}_{j_d}) \|^2_{\mathcal{H}}. \tag{11}$$

Then we average the MMD values across all channels:

$$\text{MMD}^2(i, j|\mathbf{x}) = \frac{1}{C} \sum_{c=1}^{C} \text{MMD}^2(\mathbf{P}_c(\mathbf{x}_i), \mathbf{P}_c(\mathbf{x}_j)). \tag{12}$$

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

Table 4: Performance comparison of SPACE without/with kernel embedding (*k.e.*) on CIFAR-10-C with SNN-VGG9 model regarding **Accuracy** (%). The **bold** number indicates the best result.

Method	Noise		Blur					Weather				Digital				Average
	Gauss.	Shot	Impl.	Defoc.	Glass	Motion	Zoom	Snow	Fog	Frost	Brit.	Contr.	Elas.	Pix.	JPEG	Acc.
No Adapt	72.38	74.70	58.57	63.05	63.96	64.44	71.33	76.32	43.57	75.72	82.44	22.54	75.01	70.25	84.28	66.57
SPACE	**77.98**	**79.34**	**69.41**	**71.59**	**67.76**	72.14	**74.67**	78.43	**52.80**	79.59	83.22	**23.85**	**75.49**	**76.24**	82.88	**71.03**
SPACE + *k.e.*	77.85	79.34	69.25	71.42	67.64	**72.21**	74.61	**78.48**	52.46	**79.66**	**83.25**	23.40	75.42	76.13	**82.98**	70.94

Finally, we integrate the average MMD values into the original loss function $\mathcal{L}$:

$$\mathcal{L}^*(\theta; \mathbf{x}) \triangleq \mathcal{L} + \lambda_{\text{MMD}} \sum_{1 \leq j < i \leq M} \text{MMD}^2(i, j|\mathbf{x}) \tag{13}$$

To evaluate the impact of incorporating kernel embedding into the loss function, we tested its performance on CIFAR-10-C dataset, using SNN-VGG9 model, as shown in Table 4. The results indicate that adding kernel embedding provides little to no improvement over the original SPACE method, with the average performance remaining nearly identical. This limited effectiveness can be attributed to several factors. First, the distribution differences introduced by augmentations, such as noise, blur, and weather effects, are relatively small, reducing the need for advanced alignment mechanisms like kernel embedding. Additionally, the existing similarity measures in SPACE already effectively capture the relationships between augmented samples, leaving little room for further optimization. Moreover, the added computational complexity of kernel embedding, especially the cost of kernel calculations, may even slightly hinder performance in scenarios where the original approach is already sufficient. These factors combined suggest that kernel embedding is not particularly beneficial in this experimental setting.

B Exploration of Alternative Alignment Objectives

B.1 Two Alternative Alignment Objectives

Average Membrane Potential (SPACE + *a.m.p.*) In the main paper, we use feature maps based on the spike counts over the entire time window as the alignment objective. In this section, we explore an alternative biologically inspired metric: the average membrane potential obtained from the LIF neurons in the final layer of the feature extractor. Specifically, for a given test point $\mathbf{x}_i$, the membrane potentials are captured in the feature map $\mathbf{U}(\mathbf{x}_i) \in \mathbb{R}^{T \times C \times D}$, where T, C, and D represent the time steps, the number of channels, and the spatial dimensions, respectively. For LIF neurons that spike at a specific time step, we use the membrane potential after the LIF reset, which indirectly captures the spiking information. To obtain the primary objective feature map $\mathbf{F}(\mathbf{x}_i)$, we compute the temporal average of $\mathbf{U}(\mathbf{x}_i)$ along the first dimension. The resulting feature map is then used to calculate the loss following Equations 4 and Equation 5 in the main paper. Finally, the model is updated via SGD.

Spikes through All Time Steps (SPACE + *s.t.t.*) Instead of summing the spike counts over the temporal dimension, we explicitly retain the temporal information, resulting in a feature map with an additional dimension that captures the spiking activity at each time step. This approach preserves the precise temporal dynamics of the spikes and leverages them directly for alignment. Specifically, we align the channel-wise spiking activity across all time steps between each pair of augmented samples. The update rules remain consistent with those described in the main paper.

B.2 Experimental Results and Analysis

We implement the two approaches above and evaluate on CIFAR-10-C dataset with SNN-VGG9 model. The results are shown in Table 5.

Compared to No Adapt, SPACE + *a.m.p.* improves the generalization ability of the model. However, its performance is noticeably inferior to SPACE on the majority of corruptions. This may be due to the fact that the membrane potentials of the same sample under different augmentations tend to exhibit similar patterns, making them less effective in capturing the temporal distribution characteristics of SNNs. For example, in extreme cases where the membrane potential of a LIF neuron stays near the firing threshold throughout all time steps without crossing it, SPACE + *a.m.p.* fails to capture this

Table 5: Performance comparison of SPACE with different objectives on CIFAR-10-C with SNN-VGG9 model regarding **Accuracy** (%). The **bold** number indicates the best result. Average membrane potential and spike through all time steps are referred as *a.m.p.* and *s.t.t.* respectively.

Method	Noise		Blur					Weather				Digital				Average
	Gauss.	Shot	Impl.	Defoc.	Glass	Motion	Zoom	Snow	Fog	Frost	Brit.	Contr.	Elas.	Pix.	JPEG	Acc.
No Adapt	72.38	74.70	58.57	63.05	63.96	64.44	71.33	76.32	43.57	75.72	82.44	22.54	75.01	70.25	84.28	66.57
SPACE	**77.98**	**79.34**	**69.41**	**71.59**	**67.76**	**72.14**	**74.67**	**78.43**	**52.80**	**79.59**	**83.22**	**23.85**	75.49	**76.24**	82.88	**71.03**
SPACE + *a.m.p.*	74.87	76.05	61.64	64.90	65.79	65.88	72.22	77.66	45.66	76.64	82.81	21.57	**75.53**	70.91	**84.74**	67.79
SPACE + *s.t.t.*	72.74	73.84	57.86	62.91	63.66	64.33	70.94	76.35	43.16	74.77	82.30	21.20	74.24	69.35	83.97	66.11

information. In contrast, SPACE can reflect this behavior through spike counts, which more directly encode spiking activity.

SPACE + *s.t.t.* fails to improve the generalization ability of the model. We hypothesize that this is due to the model overfitting during the alignment process. Specifically, the updates made to minimize the alignment loss may inadvertently lead to excessive specialization to the augmented samples, thereby hindering its ability to generalize effectively to unseen corruptions.

C Additional Ablation Study

Effect of Augmentation Quantity per Test Point In the main paper, we follow MEMO [13] to determine the number of augmentations M. To analyze the effect of M on our method, we evaluated the average accuracy and the evaluation time per test point the same computing resource conditions on CIFAR-10-C using SNN-VGG9 with varying augmentation quantities $M = \{2, 4, 8, 16, 32, 64, 128\}$. As shown in Table 6, at least 8 augmentations are required to achieve noticeable performance improvement. For $M = \{32, 64, 128\}$, while the performance continues to improve gradually, the evaluation time increases significantly. Considering the trade-off between accuracy and evaluation time, $M = 32$ is the optimal choice.

Table 6: Performance comparison of SPACE with different augmentation quantities on CIFAR-10-C with SNN-VGG9 model regarding **Accuracy** (%) and **Evaluation Time** (seconds) per test point.

	No Adapt	Number of Augmentation						
		2	4	8	16	32	64	128
Accuracy	66.57	54.60	66.91	69.77	70.60	71.03	71.07	71.11
Evaluation Time	0.144	0.318	0.320	0.327	0.339	0.345	0.443	0.788

D Additional Experimental Information

D.1 More Details on Datasets

In this paper, we conducted six static datasets and one neuromorphic dataset to evaluate the out-of-distribution generalization ability. They are CIFAR-10-C, CIFAR-100-C, Tiny-ImageNet-C [3], ImageNet-V2 [11], ImageNet-R [4], ImageNet-A [5] and DVS Gesture-C [6].

CIFAR-10-C, CIFAR-100-C, and Tiny-ImageNet-C These three datasets are commonly used benchmarks for evaluating the robustness of models under various types of input corruptions. They are derived from the original CIFAR-10, CIFAR-100, and Tiny-ImageNet datasets by applying 15 corruption types (*e.g.*, noise, blur, weather, and digital distortions) at 5 severity levels to the original CIFAR-10, CIFAR-100 [8], and Tiny-ImageNet [2] datasets, respectively. CIFAR-10-C and CIFAR-100-C consist of 10 and 100 classes, each with 50,000 training images and 10,000 testing images. Tiny-ImageNet-C, derived from Tiny-ImageNet, includes 200 classes with 100,000 training images and 10,000 validation images.

ImageNet V2/R/A ImageNet-V2, ImageNet-R, and ImageNet-A are three datasets designed to evaluate model robustness under different real-world challenges. ImageNet-V2 consists of a new

set of images collected under similar conditions as the original ImageNet [2], serving as a test of distribution shift. ImageNet-R focuses on image classification robustness across various artistic renditions, such as paintings, sketches, and cartoons, covering 200 ImageNet classes. ImageNet-A, on the other hand, is an adversarially filtered subset of natural images that models often misclassify, targeting their inherent weaknesses. Compared to corruption-based datasets, these benchmarks provide a better reflection of real-world scenarios and challenges.

DVS Gesture-C DVS Gesture-C is a corrupted variant of the standard DVS Gesture [1] dataset. This variant introduces six distinct corruption types: DropPixel, DropEvent, RefractoryPeriod, TimeJitter, SpatialJitter, and UniformNoise. These corruptions, implemented via the Tonic API [10], effectively simulate real-world imperfections in event-based data, including sensor noise and timing inaccuracies.

D.2 More Details on Pretrained Models

SNN-VGG with BNTT We adopt SNN-VGG with Batch-Normalization Through Time (BNTT) [7] as our main backbone model. BNTT decouples the parameters along the time axis within each layer, effectively capturing the temporal dynamics of spikes. The temporally evolving learnable parameters in BNTT enable neurons to regulate their spike rates across different time steps, facilitating low-latency and low-energy training from scratch. The pre-trained model achieves accuracies of 90.61%, 69.50% and 58.36% on the original CIFAR-10, CIFAR-100 and Tiny-ImageNet test sets respectively.

SNN-ResNet To explore other CNN architecture, we use SNN-ResNet11 [9], which trains SNNs directly via surrogate gradient. The pre-trained model used in our experiments can achieve 90.95% accuracy on the original CIFAR-10 test sets.

Spike-driven Transformer V3 To evaluate the effectiveness of our work on more complex networks and datasets, we utilized a 19M parameter model in Spike-driven Transformer V3 [12], which achieves 79.60% accuracy on the ImageNet-1K validation set. This model has a patch size of 14×14, with an embedding dimension of 360 output by the final feature extractor. Instead of using the class token, the entire embedding feature map is employed as the input to the classifier. This design naturally aligns with our method, where the channel dimension C corresponds to the embedding dimension, and the spatial dimension D matches the patch size.

SNN-ConvLSTM DVS Gesture is a neuromorphic dataset with a temporal dimension, making it well-suited for evaluating RNN-based architectures. To validate the effectiveness of our work on such networks, we employed a spike-driven 2ConvLSTM-1FC model, with hidden layer dimensions of 32 and 64, respectively. This network achieves an accuracy of 93.36% on the original DVS Gesture test set.

D.3 More Details on Implementation

To better reflect real-world scenarios, we adapt our method to independently perform adaptation on each individual test input. In the same experiment, the learning rate remains fixed across different shifted domains, as the domain of a given test sample is typically unknown in real-world settings. For each set of experiments, we employ the SGD optimizer and evaluate learning rates ranging from 0.001 to 0.8, selecting the optimal value based on performance. We trained and tested all models and datasets in 4 NVIDIA GeForce RTX 3090 GPUs.

References

[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] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[3] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

[4] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8340–8349, 2021.

[5] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *CVPR*, 2021.

[6] Yifan Huang, Wei Fang, Zhengyu Ma, Guoqi Li, and Yonghong Tian. Flexible and scalable deep dendritic spiking neural networks with multiple nonlinear branching. *arXiv preprint arXiv:2412.06355*, 2024.

[7] Youngeun Kim and Priyadarshini Panda. Revisiting batch normalization for training low-latency deep spiking neural networks from scratch. *Frontiers in neuroscience*, 15:773954, 2021.

[8] A Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Tront*, 2009.

[9] Chankyu Lee, Syed Shakib Sarwar, Priyadarshini Panda, Gopalakrishnan Srinivasan, and Kaushik Roy. Enabling spike-based backpropagation for training deep neural network architectures. *Frontiers in neuroscience*, 14:497482, 2020.

[10] Gregor Lenz, Kenneth Chaney, Sumit Bam Shrestha, Omar Oubari, Serge Picaud, and Guido Zarrella. Tonic: event-based datasets and transformations., July 2021. URL `https://doi.org/10.5281/zenodo.5079802`. Documentation available under https://tonic.readthedocs.io.

[11] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pages 5389–5400. PMLR, 2019.

[12] Man Yao, Xuerui Qiu, Tianxiang Hu, Jiakui Hu, Yuhong Chou, Keyu Tian, Jianxing Liao, Luziwei Leng, Bo Xu, and Guoqi Li. Scaling spike-driven transformer with efficient spike firing approximation training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[13] Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. *Advances in neural information processing systems*, 35:38629–38642, 2022.