# Decoupling the Contributions of Spatio-Temporal Coding: From ANNs to SNNs

**Yihao Li**[1,2], **Hanle Zheng**[1], **Jiaxin Guo**[1], **Lei Deng**[1*]
[1]Center for Brain Inspired Computing Research (CBICR),
Department of Precision Instrument, Tsinghua University
[2]Department of Computer and Information Science, University of Pennsylvania
liyihao@seas.upenn.edu, zhl22@mails.tsinghua.edu.cn,
jx-guo21@mails.tsinghua.edu.cn, leideng@mail.tsinghua.edu.cn
*Corresponding author

## Abstract

Artificial Neural Networks (ANNs) and Spiking Neural Networks (SNNs) represent two distinct but complementary approaches to information processing. ANNs, with their continuous activation functions, have been widely successful in tasks requiring nonlinear mapping, while SNNs provide a biologically plausible and energy-efficient alternative through their discrete, spike-based activity and spatio-temporal dynamics. To compare their coding schemes in-depth, we seek to decouple and analyze the contributions of spatio-temporal coding in these models. We introduce a novel mutual-information-based measure, the Exploitation Rate (ER), to quantify how information is distributed across spatial, temporal and activation domains. We also propose an incremental framework to analyze the transition between the two network paradigms. Our findings highlight the advantage of SNNs in leveraging rich temporal dynamics to compensate for their reduced complexity in activation values.

## 1 Introduction

Neural networks have progressed from early Perceptrons (1) to more advanced frameworks like Artificial Neural Networks (ANNs) and Spiking Neural Networks (SNNs) (2). Artificial Neural Networks (ANNs) (3), often referred to as the second generation of neural networks, introduced continuous nonlinearities through activation functions like sigmoid and ReLU, enabling the scalable application of deep architectures to complex tasks. ANNs have excelled in areas such as image recognition, speech processing, and natural language understanding (4; 5; 6; 7). In contrast, Spiking Neural Networks (SNNs), the third generation, process information through discrete spike-based signals, emphasizing energy-efficient computation for spatio-temporal data (8; 2; 9).

The key difference between SNNs and ANNs lies in their modes of information representation: SNNs rely on binary spikes with temporal dependencies, whereas ANNs use continuous activations. While binarizing outputs might appear to result in significant information loss, SNNs often achieve comparable performance to ANNs. One hypothesis is that SNNs' spatio-temporal coding offers an expanded capacity for information representation (10), potentially offsetting the reduced information due to discrete spikes. To test this hypothesis, we seek to decouple the contributions of spatial, temporal, and activation domains to better understand how information is encoded in these networks.

Our work investigates how ANNs and SNNs differ in their approaches to encoding information. We achieve this by introducing a novel mutual-information-based measure, the Exploitation Rate (ER), to assess the effectiveness of information representation across spatial, temporal and activation domains.

$1^{st}$ Workshop on NeuroAI @ 38th Conference on Neural Information Processing Systems (NeurIPS 2024).

We also propose an incremental framework to create network variants that bridge the gap between ANNs and SNNs, identifying key factors that enhance SNNs' spatio-temporal coding advantages.

## 2 Background

Artificial Neural Networks (ANNs) can be expressed as:

$$\mathbf{x}^{(l)} = f\left(\mathbf{W}^{(l)}\mathbf{x}^{(l-1)} + \mathbf{b}^{(l)}\right) \tag{1}$$

where $\mathbf{x}^{(l)}$ represents the activations at layer $l$, $\mathbf{W}^{(l)}$ is the weight matrix, and $\mathbf{b}^{(l)}$ is the bias term. The function $f(\cdot)$ is a nonlinear activation function like ReLU or sigmoid.

Spiking Neural Networks (SNNs) based on the Leaky Integrate-and-Fire (LIF) neuron model can be described as:

$$\begin{cases} \tau \dfrac{du_j^{(l)}}{dt} = -u_j^{(l)} + i_j^{(l)}, \quad x_j^{(l)} = 0, \quad \text{if } u_j^{(l)} < V_{\text{th}} \\ u_j^{(l)} = V_{\text{reset}}, \quad x_j^{(l)} = 1, \quad \text{if } u_j^{(l)} \geq u_{\text{th}} \end{cases} \tag{2}$$

Here, $u_j^{(l)}$ is the membrane potential of the j-th neuron that integrates inputs over time. When it exceeds the firing threshold $u_{\text{th}}$, a spike $x_j^{(l)} = 1$ is generated, and its potential resets to $u_{\text{reset}}$.

In ANNs, information is typically encoded as continuous values across spatially arranged neurons, with each neuron's activation reflecting a continuous signal. By contrast, SNNs integrate both space and time in their coding. Each layer in an SNN encodes inputs as binary temporal sequences of spikes, generated by spiking neurons that are spatially distributed within the layer (see Fig.1).

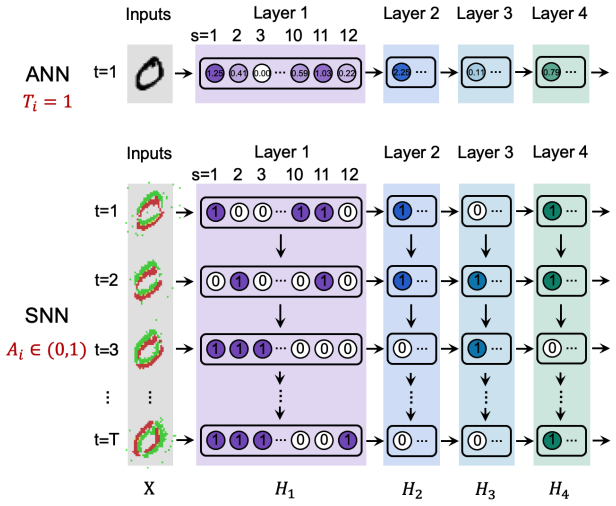



Figure 1: Comparison of coding schemes in ANNs and SNNs. Neural activities are depicted as colored circles, with activation values shown by color intensity and the numbers inside each neuron. Spatial location is indicated by neuron indices, and timing by time steps.

Building on the concept of spatio-temporal coding from neuroscience (11; 12; 13), we extend this framework to describe how both SNNs and ANNs represent information. Neural activity can be represented by a set of triples $H = \{(s_i, t_i, a_i)\}$, where $s_i$, $t_i$ and $a_i$ denotes the neuron's spatial location, the timing of each activity and the activation value, respectively. In SNNs, $a_i$ is a binary value, indicating whether a neuron fires a spike, while in ANNs, $a_i$ can take any real value. Additionally, in ANNs, $t_i$ is typically set to 1, as there is no explicit temporal component in these networks. This formulation provides a unified spatio-temporal coding scheme that applies to both network paradigms.

## 3 Methods

### 3.1 Decomposing the Information Encoded in Spatial, Temporal and Activation Domains

Neural activity within the $i$-th hidden layer can be represented by $H_i = (S_i, T_i, A_i) = (s_i, t_i, a_i)_n$. Here, $n = N \times T$, $N$ is the number of neurons within the layer and $T$ is the number of time steps. $S_i, T_i, A_i$ are the vectorized forms of $s_i, t_i, a_i$ respectively, $i = 1, 2, \ldots, n$. (see Fig. 2)

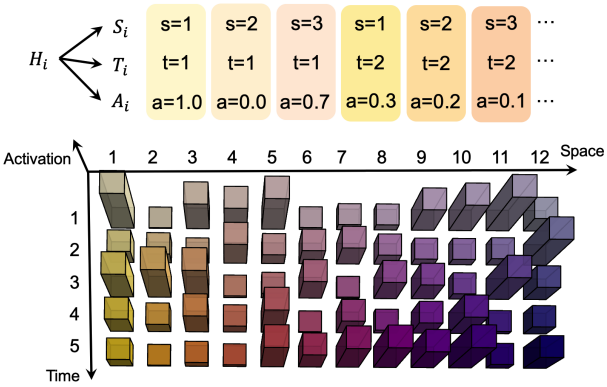



Figure 2: Representation of neural activity with spatial, temporal, and activation components $H_i = (S_i, T_i, A_i)$. The top section shows separate sequences for each component, while the bottom section presents a 3D bar plot, where the bar heights indicate the activation strengths at specific spatial locations and time points.

We use the Mutual Information Neural Estimator (MINE) (14) to quantify the mutual information between the input and hidden layer activations. MINE computes the mutual information by optimizing an upper bound via gradient descent:

$$I_\Theta(X; H) = \sup_{\theta \in \Theta} \mathbb{E}_{P_{XH}}[T_\theta] - \log(\mathbb{E}_{P_X \otimes P_H}[e^{T_\theta}]). \tag{3}$$

The function $T : \mathcal{X} \times \mathcal{H} \to \mathbb{R}$, parameterized by neural networks, distinguishes between samples from the joint distribution$P_{XH}$ (positive) and the product of the marginals$P_X \otimes P_H$ (negative).

We define mutual information between the input and hidden layer activations as $I(X; H_i) = I(X; S_i, T_i, A_i)$. We also estimate mutual information $I(X; S_i, A_i)$ and $I(X; T_i, A_i)$, by sorting along the temporal or spatial dimensions, respectively. This process removes one dimension from the representation of the neural activity. (see Supplementary Materials A.1)

We introduce the **Spatial Exploitation Rate (SER)** and **Temporal Exploitation Rate (TER)** to quantify the loss of information resulting from removing the spatial or temporal dimension:

$$\text{SER} = 1 - \frac{I(X; T_i, A_i)}{I(X; S_i, T_i, A_i)},$$

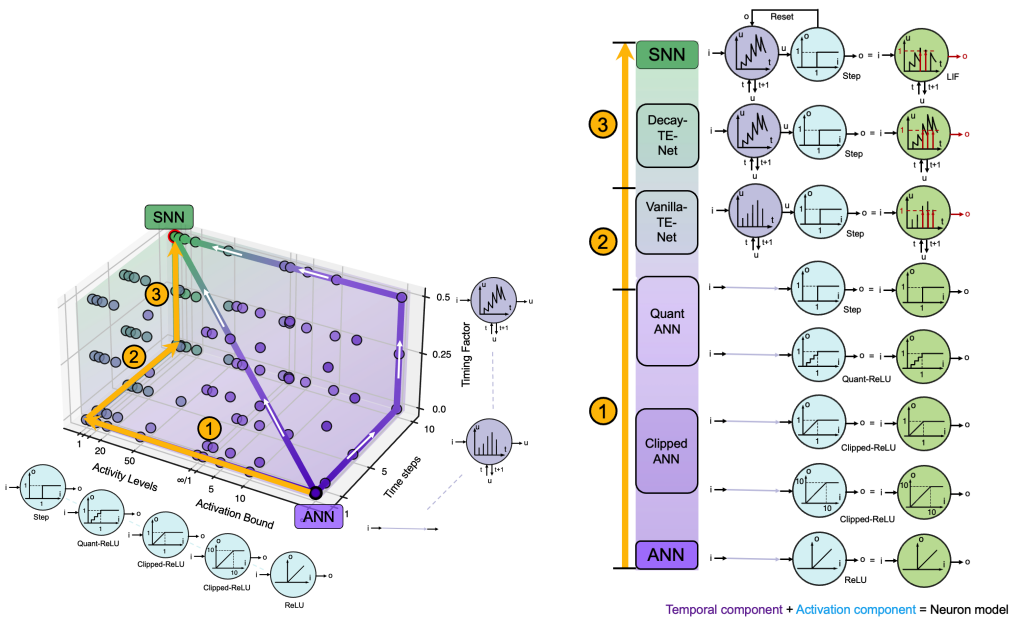
$$\text{TER} = 1 - \frac{I(X; S_i, A_i)}{I(X; S_i, T_i, A_i)}.$$

By sorting along both spatial and temporal dimensions, we compute $I(X; A_i)$. We then define the Activation Exploitation Rate (AER) as the proportion of information conveyed only by the activation values:

$$\text{AER} = \frac{I(X; A_i)}{I(X; S_i, T_i, A_i)}.$$

All measures are normalized to fall between 0 and 1, allowing for comparisons within each measure across different models but not direct comparisons between measures. Exploitation Rates (ERs) offer insights into how effectively the network utilizes spatial, temporal, and activation dimensions to encode information. Higher ER values suggest that the network is making richer use of these dimensions.

### 3.2 Constructing Intermediate Variants Evolving from ANNs to SNNs




(a) (b)

Figure 3: Schematic of intermediate variants between ANNs and SNNs. (a) A coordinate frame showing an ANN (bottom-right), an SNN (top-left), and their intermediate variants. The color gradient represents similarity: purple for ANNs and green for SNNs. (b) Detailed design of ANNs, SNNs, and their variants, featuring LIF-like temporal dynamics (optional) and time-independent activation functions.

To study the transition from ANNs to SNNs, we define intermediate models that allow us to examine the impact of key variables such as activation bounds, activation discretization, time steps, and timing factors.

Each intermediate model consists of two components: an optional temporal component that simulates how the membrane potential accumulates over time, and an activation component that applies the nonlinear activation, which is independent of the temporal dynamics. Together, these components form variants that progressively move from ANN-like behaviors to the SNN-like spiking dynamics (see Fig.3 and Supplementary Material A.2).

The transition occurs in three phases: (1) At $T = 1$, activations are clipped and quantized to simulate discrete outputs. (2) Time steps are increased, enabling temporal processing. (3) Temporal dependency is added, where membrane potential accumulates and decays, along with a reset mechanism, completing the transition to the SNN dynamics.

## 4 Results

### 4.1 Trajectory view: Transitioning from ANNs to SNNs

We compute the Exploitation Rates (ERs) including AER, SER and TER for the models along the transition path described in Section 3.2 on four datasets (see experimental setup in Supplementary Materials A.3). The plotted results demonstrate the transition of these ERs as the models evolve from ANNs to SNNs (see Fig. 4).

In Phase 1 (activation bound and discretization), as the activation function is progressively clipped and quantized, the AER decreases while SER increases. This indicates a redistribution of information from the activation dimension to the spatial dimension. By limiting the range of possible activation

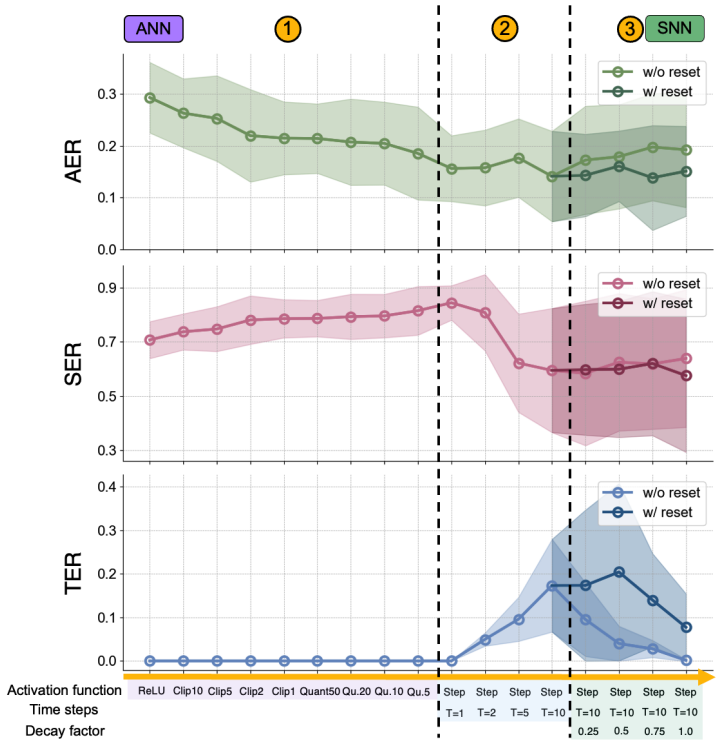



Figure 4: Activation Exploitation Rate (AER), Spatial Exploitation Rate (SER), and Temporal Exploitation Rate (TER) for models along the transition path from ANNs to SNNs.

values, the network loses some of the variability it could draw upon in the activation dimension, causing it to depend more heavily on spatial patterns to represent information.

In Phase 2 (temporal representation), the introduction of time steps leads to a large increase in the Temporal Exploitation Rate (TER), while SER decreases, suggesting a shift in information from the spatial to the temporal domain.

In Phase 3 (temporal dependency), we analyze how temporal dependency impacts ER values in Decay-TE-Nets (without reset) and SNNs (with reset) as timing factors increase.

For Decay-TE-Nets, increasing the timing factor strengthens dependencies between consecutive time steps, leading to reduced variability in temporal neural activity. This weakens the temporal dimension's role in encoding information, causing a sharp decline in TER. To compensate, the activation dimension absorbs more information, resulting in a gradual increase in AER.

SNNs mitigate this effect by introducing a reset mechanism that reduces interdependence between timesteps. At moderate timing factors, this allows SNNs to balance temporal relationships, resulting in an initial increase in TER. However, as the timing factor continues to grow, the reset mechanism struggles to fully decouple correlations, causing TER to decline. Throughout, SNNs maintain higher TER values compared to Decay-TE-Nets, while their AER remains lower and SER shows no consistent pattern.

### 4.2 Grid view: Sparse sampling in the model space

To obtain a more comprehensive view of how these three measures vary and interact across different architectures, we sparsely sample network variants from a grid and compute the corresponding AER, SER, and TER values for each model on Sequential-MNIST (Fig. 5). As expected, the observed trends along trajectories parallel to each axis align with the findings in Section 4.1, supporting our analysis of the roles of activation bounds, discretization, temporal representation, and temporal dependencies.

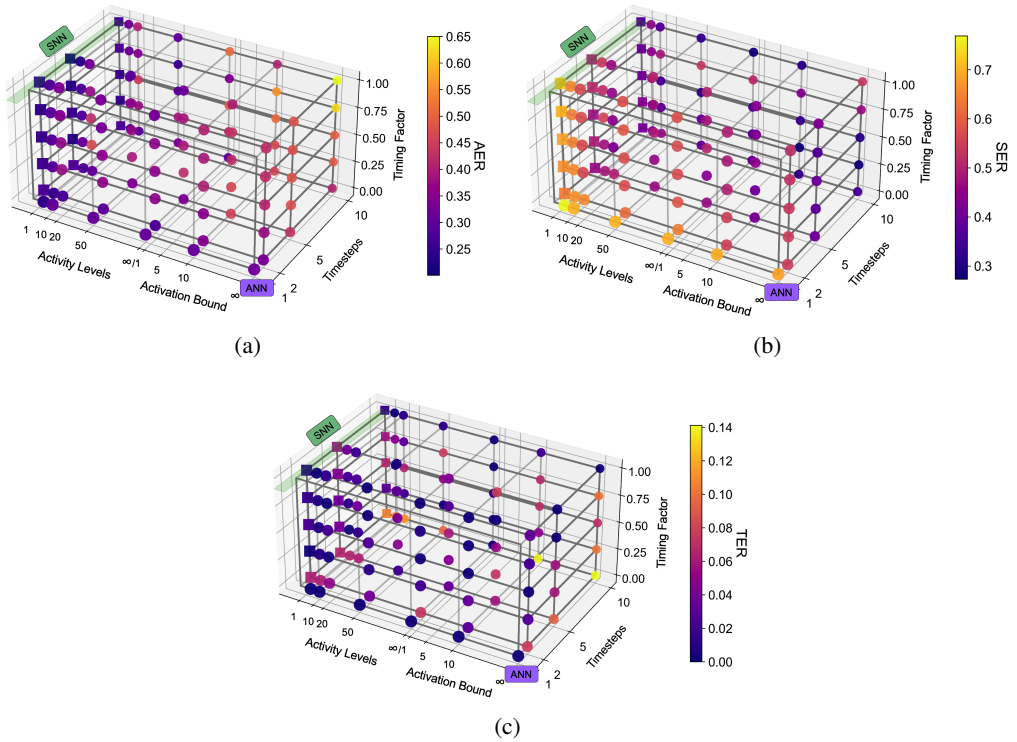



Figure 5: Visualization of ERs in model space on Sequential MNIST. Circle markers represent models without a reset mechanism, namely Decay-TE-Nets, while square markers denote models with a reset mechanism, namely SNNs.

## 5 Conclusion

In this work, we introduced a novel measure to decouple and assess the contributions of spatial, temporal, and activation coding in neural information processing. Our method provides a systematic framework for evaluating how information is encoded across these dimensions in ANNs, SNNs and their variants.

Through our analysis of the transition from ANNs to SNNs, we observed a gradual shift in the relative importance of these coding strategies. Specifically, the contribution of activation coding diminishes as spiking behavior emerges, while the role of temporal coding becomes increasingly significant. This highlights the unique advantage of SNNs in utilizing rich temporal dynamics to process information effectively, compensating for the reduced complexity in activation values.

More specifically, activation bound and discretization reduce variability in neural activations, shifting information to the spatial domain. An increase in time steps redistributes information into the temporal dimension. Temporal dependency, when combined with the reset mechanism, optimizes temporal information encoding by balancing memory retention and partitioning across time steps. These factors provide new understanding of how neural networks balance and optimize information encoding across the three domains.

## 6 Acknowledgements

This work was partially supported by the National Natural Science Foundation of China (No. 62276151) and Chinese Institute for Brain Research, Beijing.

## References


[1] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[2] K. Roy, A. Jaiswal, and P. Panda, "Towards spike-based machine intelligence with neuromorphic computing," *Nature*, vol. 575, no. 7784, pp. 607–617, 2019.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[5] Y. Miao, M. Gowayyed, and F. Metze, "Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding," in *2015 IEEE workshop on automatic speech recognition and understanding (ASRU)*. IEEE, 2015, pp. 167–174.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[7] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of cnn and rnn for natural language processing," *arXiv preprint arXiv:1702.01923*, 2017.

[8] E. Hunsberger and C. Eliasmith, "Training spiking deep networks for neuromorphic hardware," *arXiv preprint arXiv:1611.05141*, 2016.

[9] L. Deng, Y. Wu, X. Hu, L. Liang, Y. Ding, G. Li, G. Zhao, P. Li, and Y. Xie, "Rethinking the performance comparison between snns and anns," *Neural networks*, vol. 121, pp. 294–307, 2020.

[10] R. Rubin, R. Monasson, and H. Sompolinsky, "Theory of spike timing-based neural classifiers," *Physical review letters*, vol. 105, no. 21, p. 218102, 2010.

[11] J. W. Pillow, J. Shlens, L. Paninski, A. Sher, A. M. Litke, E. Chichilnisky, and E. P. Simoncelli, "Spatio-temporal correlations and visual signalling in a complete neuronal population," *Nature*, vol. 454, no. 7207, pp. 995–999, 2008.

[12] E. Chong and D. Rinberg, "Behavioral readout of spatio-temporal codes in olfaction," *Current Opinion in Neurobiology*, vol. 52, pp. 18–24, 2018, systems Neuroscience. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0959438818300138

[13] W. Xie, J. H. Wittig Jr, J. I. Chapeton, M. El-Kalliny, S. N. Jackson, S. K. Inati, and K. A. Zaghloul, "Neuronal sequences in population bursts encode information in human cortex," *Nature*, pp. 1–8, 2024.

[14] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm, "Mutual information neural estimation," in *International conference on machine learning*. PMLR, 2018, pp. 531–540.

[15] W. He, Y. Wu, L. Deng, G. Li, H. Wang, Y. Tian, W. Ding, W. Wang, and Y. Xie, "Comparing snns and rnns on neuromorphic vision datasets: Similarities and differences," *Neural Networks*, vol. 132, pp. 108–120, 2020.


## A Appendix / Supplemental Materials

### A.1 Method: Shuffling for Partial Hidden Layer Activation

In terms of $I(X; S_i, A_i)$, $(S_i, A_i)$ only represents the spatial location and activation value of each neural activity. The activation $a_i$s of each neuron over $T$ time steps are determined; however,

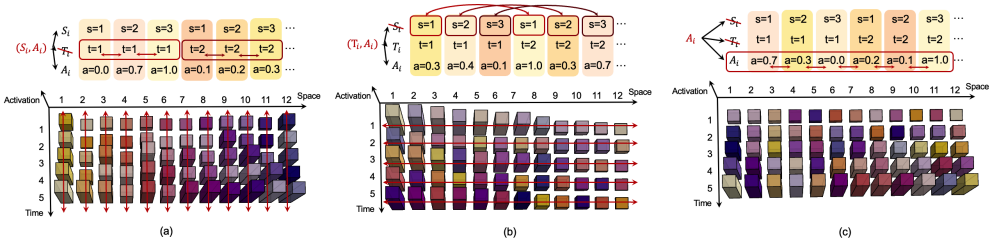



Figure 6: (a) Estimate $I(X; S_i, A_i)$: sort the neural activity along the temporal dimension. (b) Estimate $I(X; T_i, A_i)$: sort the neural activity along the temporal dimension. (c) Estimate $I(X; A_i)$: sort the neural activity along both dimensions.

the sequence of these activations $a_i$s remains unknown. To isolate the influence of the temporal dimension, we sort the activities of each neuron on this dimension respectively. (see Fig. 6(a))

Similarly, when it comes to $I(X; T_i, A_i)$, $(T_i, A_i)$ only represents the timing and activation value of each neural activity. The activation $a_i$s of each time over $N$ neurons are determined; however, the sequence of these activations $a_i$s remains unknown. To isolate the influence of the spatial dimension, we randomly shuffle the activities of each neuron on this dimension respectively. (see Fig.6(b))

If we sort them along both the temporal and the spatial dimensions (see Fig.6(c)), we immediately get $I(X; A_i)$, signifying the fraction of information the activation values themselves carry.

### A.2 Method: Details on Constructing Intermediate Models

Both ANNs and SNNs are incorporated within the same NN framework but can be distinguished by several discriminative variables that can be identified and isolated. By isolating these variables, we can gradually manipulate them to simulate the evolution from ANNs to SNNs. In doing so, we construct several intermediate variants that interpolate such evolution, thus enabling comparative studies to investigate the functional differences between ANNs and SNNs.

#### A.2.1 Identifying the Comparative Dimensions and Discriminative Variables

We propose three key comparative dimensions that differentiate SNNs from ANNs: activation bound and discretization, temporal representation, and temporal dependency. We then identify the discriminative variables that characterize each comparative dimension.

**Activation Bound and Discretization.** The difference between ANNs and SNNs in their activation patterns, i.e., continuous versus binary activation, is influenced by two stages: bounding and discretization. Initially, the $ReLU$ activation function in ANNs is clipped to a decreasing range, until $(0, 1)$. Once bounded, the continuous activation can be discretized with decreasing granularity (from infinite to singular levels), converging towards the binary activation of SNNs, i.e., the spike event. Consequently, we identify two variables: ***activation bound*** and ***activation levels***. The former determines the range of the activation function, and the latter signifies the number of discretization levels within that range. To concretize these concepts, below are the formulas for Clipped ReLU and Quant ReLU:

- Clipped ReLU:
  Clipped ReLU is a variant of ReLU where the activation values are restricted within a specific range. Here, we set the range to [0, $\alpha$], and the formula is as follows:

$$\text{Clipped ReLU}(x) = \min(\max(x, 0), \alpha) \tag{4}$$

  where $\alpha$ represents the upper limit of the clipping. When the input $x$ is less than 0, the output is 0; when the input $x$ is greater than $\alpha$, the output is $\alpha$; otherwise, the output is equal to the input value.

- Quant ReLU:
  Quant ReLU further discretizes the Clipped ReLU by restricting the activation values to discrete quantization levels. For example, we can discretize it into $n$ levels, and the formula

is as follows:

$$\text{Quant ReLU}(x) = \frac{1}{n} \lfloor n \cdot \text{Clipped ReLU}(x) \rfloor \tag{5}$$

where $\lfloor \cdot \rfloor$ denotes the floor operation, and $n$ represents the number of quantization levels. In this way, continuous activation values are quantized into $n$ discrete levels.

These formulas help us better understand the differences in activation patterns between ANNs and SNNs. By limiting the range of the activation function and performing quantization, we can simulate the gradual transition from ANNs to SNNs, thereby achieving a spiking representation of neural networks.

**Temporal Representation.** Compared with typical ANNs, SNNs possess an additional temporal domain for information representation. Each neuron in an SNN receive inputs from multiple time steps, that is

$$\tau \frac{du}{dt} = -u + i, \tag{6}$$

where $u$ represents the membrane potential, $\tau$ denotes the time constant, and $i$ correspond to the input. Thus, the information flow can be unfolded over time. Consequently, a temporal dimension is introduced into the code for the corresponding input. To quantify the temporal expansion of encoded representation, we will use the time steps of the network as a metric for reflecting the degree of temporal domain expansion.In general, shorter time steps can conserve computational resources, while longer time steps enable more precise tracking of dynamic changes in input signals, allowing the network to capture more complex temporal patterns.

**Temporal Dependency.** Another key difference between ANNs and SNNs lies in their neural dynamics. In SNNs, the decay of the membrane potential and the reset mechanism create a temporal dependency in spiking neurons. In contrast, the neural models in ANNs are simple non-linear functions that lack any inherent dynamics. For the LIF neuron model in SNNs, the update of the membrane potential can be written as:

$$\begin{cases} u = u_{reset}, \text{if } u \geq V_{th} \text{ and } \textbf{\textit{allow reset}} == \textbf{True} \\ \tau \dfrac{du}{dt} = -u + i, \text{else} \end{cases} \tag{7}$$

The ***timing factor***, which is expressed as $d = 1 - \frac{1}{\tau}$ (where $\tau$ is the time constant in the LIF neuron model in Equation 2), characterizes the decaying behavior of the membrane potential with time. The timing factor can be interpreted as the decaying coefficient that the membrane potential is multiplied by at each iteration. A larger timing factor results in a slower decay of the membrane potential in the absence of input stimuli. Conversely, a smaller timing factor causes the membrane potential to decay more quickly, as it approaches its steady-state value more rapidly. In addition, the occurrence of the reset mechanism can be represented by a Boolean parameter, ***allow reset***.

### A.2.2 Constructing Intermediate Variants

With the identified discriminative variables (activation bounds, activation levels, timesteps, and timing factors) as axes, we can plot both ANNs and SNNs within a coordinate framework (see Fig. 3(a)). The trajectories that connect ANNs to SNNs simulate the evolution between these models, marking each intermediate model as distinct points along this continuum. To elucidate the role of each variable in differentiating ANNs from SNNs, we emphasize analyzing the evolutionary trajectories parallel to the axes. This means constructing comparative models with one or more variables controlled.

Detailed designs of the intermediate variants are displayed in Fig. 3(b). To study the roles of activation bound and discretization, we first clip the commonly used $ReLU$ function in ANNs to the defined *activation bound*. These clipped functions (referred to as $Clipped-ReLU$) are then discretized into various levels, forming the $Quantized-ReLU$ (i.e., $Quant-ReLU$), termed as Clipped ANN variant and Quant ANN variants, respectively. As the quantization level is equal to 1, the activation function of the responding Quant ANN variant is acutally the Heaviside step function, which generates binary values like spike events. These different activation functions form the activation component of a neuron model.

Next, we consider temporal representation and temporal dependency. Temporally-Expanded Networks (TE-Nets) are constructed by introducing the temporal domain to the Quant ANN variants,

which expand the temporal representation of each neuron to several timesteps. Intermediate models with varying timing factors, with or without the reset mechanism, are also created to explore the role of temporal dependency. Specifically, **Vanilla-TE-Net** represents the baseline model, where each neuron's state is fully reset at every timestep, resulting in no memory of previous states, as shown in Fig 3(b). The update of its neuron model can be written as

$$\begin{cases} u(t) = i(t), \\ o(t) = 1, \text{if } u(t) \geq V_{th};\ o(t) = 0, else. \end{cases} \tag{8}$$

**Decay-TE-Net**, on the other hand, incorporates a timing factor $d$ into the membrane potential, allowing for gradual fading of past states over time. Its neuron model can be described by the following equation:

$$\begin{cases} u(t) = d \cdot u(t-1) + i(t), \\ o(t) = 1, \text{if } u(t) \geq V_{th};\ o(t) = 0, else. \end{cases} \tag{9}$$

This decay mechanism introduces a controlled form of the temporal memory. When a reset mechanism is introduced to Decay-TE-Net, it transforms into an SNN. In this case, if the membrane potential of a neuron exceeds a predefined threshold, it is reset to zero, mimicking the spiking behavior of biological neurons. The update of the membrane potential can be represented as Equation (**??**) shows. By combining the temporal component and the activation component, we can construct intermediate variants of ANNs and SNNs, enabling us to systematically study the evolution from ANNs to SNNs.

### A.3 Experimental Settings

**Datasets:** We evaluated our method on four datasets: MNIST, Fashion-MNIST, N-MNIST, and DVS128 Gesture. MNIST and Fashion-MNIST, traditionally static image recognition tasks, were reformulated as sequence learning tasks. Each 28 × 28 image was transformed into a sequence of T timesteps by dividing the image into vertical slices, following the Sequential MNIST(**?** )/Fashion-MNIST approach. For the dynamic datasets, N-MNIST and DVS128 Gesture, event streams were partitioned into T frames. In both static and dynamic cases, these frame sequences fully represented the original images or event streams.

**Network Structure and Training Details**:

In each task, we utilize one main network $\theta$ for the original object recognition task and four additional MINE networks $\theta_{\text{MINE}}$ to estimate different mutual information. The main network consists of four feedforward layers: two convolutional layers for MNIST, Fashion-MNIST, and N-MNIST, or four convolutional layers for DVS-Gesture, followed by two fully connected layers. The task-specific loss, denoted as $L_{task}$, is the mean squared error between the averaged output (firing rate of the output layer) and the one-hot label, a strategy backed by prior research (15) for fair comparative studies between ANNs and SNNs. The four distinct MINE networks are employed to estimate $I(X; A, S, T)$, $I(X; A, S)$, $I(X; A, T)$, and $I(X; A)$, respectively, for the output of the first fully connected layer of the network $\theta$. Each MINE networks is a fully-connected two-layered LSTM network with a single output to estimate the $T_\theta$ function in the Donsker-Varadhan upper bound:

$$I_\Theta(X; H) = \sup_{\theta \in \Theta} \mathbb{E}_{P_{XH}}[T_\theta] - log(\mathbb{E}_{P_H \otimes P_H}[e^{T_\theta}]).$$