LEARNING ROBUST EVENT REPRESENTATIONS THROUGH FINITE-TIME DYNAMICAL STABILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Event cameras are a class of bio-inspired vision sensors with advantages such as high temporal resolution, low latency, and high dynamic range. Existing state-of-the-art event processing methods mainly rely on deep networks to learn features, but these methods do not consider the internal temporal dynamics of the model, causing models to learn fragile temporal features and thus suffer significant performance degradation under distribution shifts such as event dropout, temporal jitter, and polarity perturbations. To address this issue, incorporating recent findings in the primary visual cortex, we propose Lyapunov-Regularized Dynamical Representation Learning (LRDR) for event information processing. Specifically, this method maps event streams into a trainable nonlinear dynamical system and uses finite-time Lyapunov exponents to constrain the dynamical stability of different samples, encouraging the same sample to maintain consistency under perturbations while enhancing dynamical separability between different classes. Across four event-recognition benchmarks, LRDR achieves state-of-the-art classification accuracy of 85.7% and 99.9% on N-Caltech101 and ASL-DVS, respectively. Our results demonstrate that finite-time stability is not merely an auxiliary property for analyzing the dynamics of neural networks after training, but can be directly optimized as a learnable representational geometry, thereby providing more stable and robust temporal representations for event-based vision models.

1 INTRODUCTION

Event cameras are an emerging class of neuromorphic vision sensors that fundamentally differ from the frame-based sampling mechanism of conventional cameras. Their event signals encode spatiotemporal information, including timestamps, spatial coordinates, and polarity; they sparsely record brightness changes as quadruples (x, y, p, t) and, with microsecond-level temporal resolution and ultra-high dynamic range, are ideally suited for challenging scenarios involving high-speed motion and complex illumination (Gallego et al., 2022). However, because each pixel of an event camera operates independently, this unique asynchronous operation makes the quality of event representations highly sensitive to event sparsity, polarity distribution, temporal jitter, and sensor noise, often leading to insufficient overall model stability. To address this issue, previous works have converted raw event streams into various structured representations, such as 2D frame images, event surfaces, voxel grids, graph structures, and Transformer features, to adapt them to deep learning models (Lagorce et al., 2017; Gehrig et al., 2019; Cannici et al., 2020; Sabater et al., 2022; Hagenaars et al., 2021).

However, existing state-of-the-art event processing methods focus their optimization on geometric distributions in feature space, treating temporal information largely as an auxiliary modeling aid, without sufficiently or explicitly exploiting the dynamical evolution characteristics inherent in event streams. Specifically, these methods are typically trained end-to-end with classification loss as the primary optimization objective, lacking explicit constraints on the stability of the model's internal temporal dynamics, which causes models to learn fragile temporal cues such as precise timestamp order and specific polarity arrangements. Under distribution shifts such as event dropout, temporal jitter, and polarity perturbations, the performance of these models often degrades significantly.

To better exploit the temporal information contained in event streams and address the limited flexibility and robustness of existing event-processing methods, we propose Lyapunov-Regularized Dynamical

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

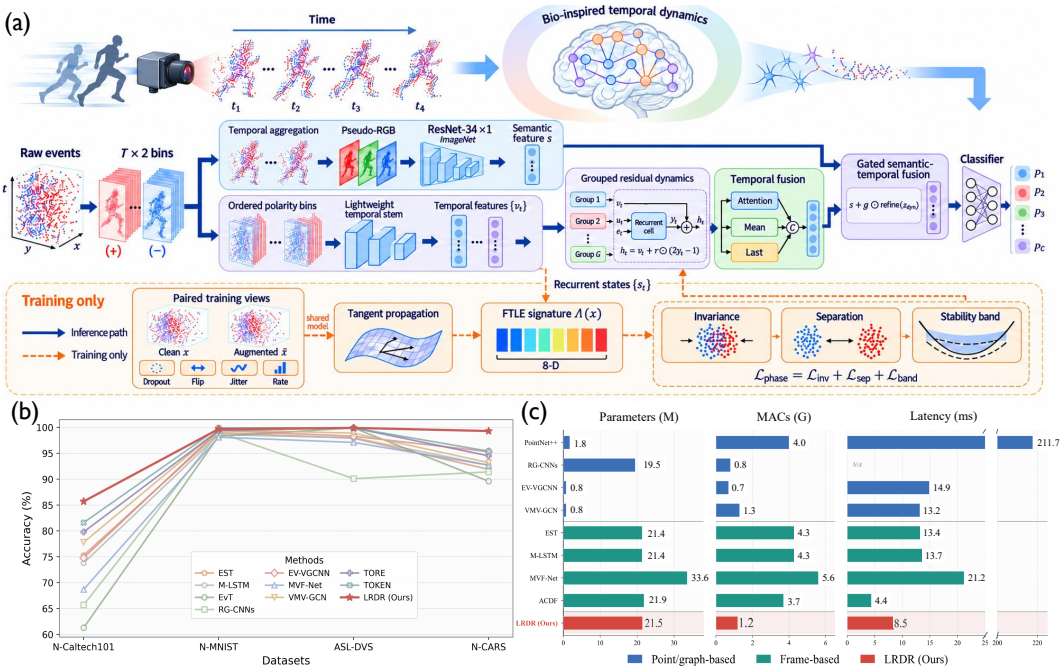


Figure 1: Overview and empirical evaluation of LRDR. (a) Event streams are divided into polarity-aware temporal bins, mapped to pseudo-RGB frames, encoded by an ImageNet-pretrained ResNet-34, and processed through grouped residual dynamics and temporal fusion. During training, paired clean and augmented views regularize the group-wise finite-time Lyapunov exponent (FTLE) signature using invariance, separation, and stability-band objectives; the orange dashed branch is removed at inference. (b) Classification accuracy on four event-recognition datasets. (c) Comparison of model parameters, MACs, and inference latency, where lower values indicate greater efficiency. LRDR achieves competitive accuracy with only 1.2 G MACs and 8.5 ms latency.

Representation Learning (LRDR). LRDR draws inspiration from the nonlinear dynamical mechanisms of the primary visual cortex and introduces learnable Continuous-Coupled Neural Networks (CCNN) dynamics for event-sequence processing. Specifically, we first divide the input into multiple temporal bins according to event polarity, preserving positive events, negative events, and overall motion information. The model then employs two complementary branches for feature extraction. One branch jointly integrates all temporal bins into a pseudo-RGB representation and performs a single forward pass through an ImageNet-pretrained ResNet-34 to extract global semantic features. The other branch preserves the complete temporal ordering of the event sequence, uses a lightweight temporal encoder to obtain features at each time step, and feeds them into a group-wise trainable CCNN dynamical module.

CCNN is a neural network capable of accurately characterizing the dynamic properties of the primary visual cortex (Liu et al., 2022). It can encode static input signals into periodic responses and periodic input signals into chaotic responses. In LRDR, CCNN is mainly used to describe the continuous temporal evolution of event features. Its output does not directly replace the original semantic features, but instead modulates the temporal representation in a residual manner. This design preserves the semantic information learned by the pretrained network while explicitly modeling the nonlinear temporal evolution of event streams. Building on this dynamical representation, we further use finite-time Lyapunov exponents (FTLE) to characterize the sensitivity of the resulting trajectories to perturbations. During training, we encourage the same sample to maintain similar dynamical characteristics before and after perturbation, while simultaneously enlarging the dynamical separation between samples from different classes. During training, we compute differentiable group-wise FTLE signatures via tangent vector propagation without forming full Jacobians. These signatures are regularized by augmentation invariance, label-aware phase separation, and bounded stability, and the phase loss is introduced only after the classifier has largely converged. At inference, all FTLE-related

computation is removed. LRDR achieves 85.7% on N-Caltech101 and 99.9% on ASL-DVS, while ranking among the top two methods on both N-MNIST and N-CARS.

The main contributions of this work are summarized as follows:

- We propose LRDR, a novel event representation framework that uses learnable, primary visual cortex-inspired CCNN to model the temporal evolution of event features. During training, LRDR directly regularizes the sample-specific finite-time stability of dynamics, rather than analyzing stability only after training, to learn more robust event representations.
- We further use finite-time Lyapunov exponents (FTLEs) to characterize the stability of CCNN trajectories and incorporate them into training through differentiable state–tangent propagation, encouraging perturbation consistency and inter-class dynamical separation.
- We evaluate the proposed framework on multiple event-classification datasets and under various event perturbations. The results demonstrate competitive classification performance together with good robustness and generalization.

2 Related Work

Event representations. Event-stream representations range from event counts and time surfaces to voxel grids, recurrent surfaces, graphs, Transformers, and spiking neural networks (Lagorce et al., 2017; Gehrig et al., 2019; Cannici et al., 2020; Sabater et al., 2022; Hagenaars et al., 2021). Frame-like encodings provide compatibility with conventional visual backbones, while recurrent and structured representations can preserve finer temporal organization. Despite these architectural differences, most methods are trained end-to-end with classification loss as the primary objective and lack explicit constraints on the stability of their internal temporal dynamics. As a result, they may learn fragile temporal cues, such as precise timestamp ordering and specific polarity arrangements. Under distribution shifts such as event dropout, temporal jitter, and polarity perturbations, their performance often degrades significantly.

Dynamics-inspired event processing. Chen et al. (2025) proposed a chaotic dynamics-based event processing framework that uses fixed CCNN dynamics to transform event-polarity sequences into periodic or more complex dynamic responses, followed by continuous wavelet transform (CWT) and thresholding for subsequent analysis. Although this method demonstrates the feasibility of nonlinear dynamics for event representation, its CCNN parameters are not learnable and its post-processing relies on hand-crafted design, making it difficult to adapt the dynamical representation to specific recognition tasks.

Learned dynamical systems and stability. Neural ODEs and neural controlled differential equations model sequential data through continuous input-driven state evolution (Chen et al., 2018; Kidger et al., 2020). Prior studies analyze how dynamical stability shapes information propagation and trainability in deep and recurrent networks (Poole et al., 2016; Schoenholz et al., 2017). More recently, Wöhrer & Kuehn (2026) regularize FTLEs in neural ODEs for adversarial robustness. In contrast, LRDR learns sample-conditioned group-wise stability signatures from driven event dynamics and jointly enforces perturbation invariance, inter-class separation, and bounded finite-time evolution.

Motivated by these limitations, we make the event dynamics learnable and explicitly optimize their sample-conditioned finite-time stability. Specifically, LRDR uses trainable CCNN dynamics to model temporal evolution and regularizes group-wise FTLE signatures for perturbation consistency, inter-class separation, and bounded stability, turning dynamical stability from a post-hoc analysis tool into a training objective for robust event representation learning.

3 Method

3.1 Architecture and Event Encoding

An event sample is represented as

$$\mathcal{E} = \{(x_n, y_n, p_n, t_n)\}_{n=1}^{N}. \tag{1}$$

We divide its duration into T intervals and accumulate positive- and negative-polarity events separately, producing

$$\boldsymbol{X} \in \mathbb{R}^{T\times 2\times H\times W}. \tag{2}$$

As illustrated in Figure 1, LRDR follows a dual-branch design. The semantic branch jointly aggregates all T temporal bins and performs a single ResNet-34 forward pass, while the lightweight temporal branch preserves the ordered T-bin sequence for grouped nonlinear dynamics. Event counts are compressed element-wise using $\log(1+x)$, and spatial coordinates are resized using aspect-ratio-preserving letterboxing. Denoting the compressed polarity maps by P_t and N_t, we compute

$$\bar{P} = \frac{1}{T}\sum_{t=1}^{T} P_t, \qquad \bar{N} = \frac{1}{T}\sum_{t=1}^{T} N_t, \qquad C = \mathrm{clip}(\bar{P}+\bar{N}, 0, 1), \tag{3}$$

and construct the temporally aggregated pseudo-RGB input

$$I_{\mathrm{sem}} = \mathrm{stack}\left(\bar{P}, \mathrm{clip}(\bar{P}+\bar{N}+\eta C, 0, 1), \bar{N}\right), \tag{4}$$

where $\eta = 0.35$. The semantic branch performs a single ImageNet-pretrained ResNet-34 forward pass:

$$\boldsymbol{s} = \mathrm{GELU}\left(\mathrm{LN}(W_s E_\phi(I_{\mathrm{sem}}) + b_s)\right) \in \mathbb{R}^D. \tag{5}$$

In parallel, a lightweight temporal stem preserves the ordered polarity-bin sequence:

$$X = \mathrm{stack}_{t=1}^{T}(P_t, N_t), \qquad \{\boldsymbol{v}_t\}_{t=1}^{T} = \mathcal{T}_\psi(X), \quad \boldsymbol{v}_t \in \mathbb{R}^D. \tag{6}$$

where $\boldsymbol{v}_t \in \mathbb{R}^D$ and $D = 128$. The sequence $\{\boldsymbol{v}_t\}_{t=1}^{T}$ is passed to the grouped dynamical module, while $\boldsymbol{s}$ is retained for subsequent semantic–temporal fusion.

3.2 Grouped Residual Dynamics

We partition $\boldsymbol{v}_t$ into G groups of width $c = D/G$. For each group g, the recurrent state evolves as

$$\boldsymbol{u}_t^{(g)} = \boldsymbol{a}_f^{(g)} \odot \boldsymbol{u}_{t-1}^{(g)} + W_x^{(g)} \boldsymbol{v}_t^{(g)} + \boldsymbol{b}_u^{(g)}, \tag{7}$$

$$\boldsymbol{e}_t^{(g)} = \boldsymbol{a}_e^{(g)} \odot \boldsymbol{e}_{t-1}^{(g)} + \boldsymbol{v}_e^{(g)} \odot \boldsymbol{y}_{t-1}^{(g)}, \tag{8}$$

$$\boldsymbol{y}_t^{(g)} = \sigma\Big(\boldsymbol{\beta}^{(g)} \odot (\boldsymbol{u}_t^{(g)} - \boldsymbol{e}_t^{(g)})\Big). \tag{9}$$

The parameters are constrained by

$$\boldsymbol{a}_f = \sigma(\boldsymbol{\rho}_f), \qquad \boldsymbol{a}_e = \sigma(\boldsymbol{\rho}_e), \tag{10}$$

$$\boldsymbol{v}_e = \mathrm{softplus}(\boldsymbol{\rho}_v), \qquad \boldsymbol{\beta} = \mathrm{softplus}(\boldsymbol{\rho}_\beta) + 0.25. \tag{11}$$

This update retains the feed and dynamic-threshold interaction of CCNN models (Liu et al., 2022; Chen et al., 2025), while all parameters are learned end-to-end. CCNN can characterize the dynamical properties of the primary visual cortex in the most refined way. Specifically, under static input, CCNN produces periodic output; under dynamic input, it enters a chaotic state with positive Lyapunov exponents. Detailed proof and analysis are provided in the APPENDIX B.

Because the bounded response $\boldsymbol{y}_t$ may discard information already captured by the pretrained encoder, we use it to residually modulate, rather than replace, the semantic feature:

$$\boldsymbol{h}_t^{(g)} = \boldsymbol{v}_t^{(g)} + \boldsymbol{r}^{(g)} \odot (2\boldsymbol{y}_t^{(g)} - 1), \qquad \boldsymbol{r}^{(g)} = \sigma(\boldsymbol{\rho}_r^{(g)}). \tag{12}$$

Concatenating all groups gives $\boldsymbol{h}_t = [\boldsymbol{h}_t^{(1)}; \ldots; \boldsymbol{h}_t^{(G)}] \in \mathbb{R}^D$. Thus, the identity path preserves pretrained semantics, while the recurrent path introduces data-dependent temporal modulation.

3.3 Temporal Fusion and Classification

We summarize the trajectory using attention-weighted, mean, and final features. Specifically,

$$a_t = w_2^\top \mathrm{GELU}(W_1 \mathrm{LN}(\boldsymbol{h}_t)), \qquad \alpha_t = \frac{\exp(a_t)}{\sum_j \exp(a_j)}, \tag{13}$$

and

$$\boldsymbol{z}_{\text{attn}} = \sum_t \alpha_t \boldsymbol{h}_t, \qquad \boldsymbol{z}_{\text{mean}} = \frac{1}{T}\sum_t \boldsymbol{h}_t, \qquad \boldsymbol{z}_{\text{last}} = \boldsymbol{h}_T. \tag{14}$$

The final representation is

$$\boldsymbol{z}_{\text{dyn}} = \text{GELU}\left(\text{LN}\left(W_f[\boldsymbol{z}_{\text{attn}}; \boldsymbol{z}_{\text{mean}}; \boldsymbol{z}_{\text{last}}]\right)\right), \tag{15}$$

$$\boldsymbol{g} = \sigma\left(W_g[\boldsymbol{s}; \boldsymbol{z}_{\text{dyn}}] + b_g\right), \tag{16}$$

$$\boldsymbol{z} = \boldsymbol{s} + \boldsymbol{g} \odot \left(W_r \boldsymbol{z}_{\text{dyn}} + b_r\right). \tag{17}$$

followed by a dropout MLP classifier $f(\boldsymbol{x})$. This constitutes the complete inference path. During training, we additionally regularize the stability of the recurrent trajectory.

3.4 Group-Wise Finite-Time Lyapunov Signature

Let

$$\boldsymbol{s}_t^{(g)} = [\boldsymbol{u}_t^{(g)}; \boldsymbol{e}_t^{(g)}; \boldsymbol{y}_t^{(g)}] = F_\theta^{(g)}(\boldsymbol{s}_{t-1}^{(g)}, \boldsymbol{v}_t^{(g)}) \tag{18}$$

denote the state transition of group g. Holding the current input fixed, its state Jacobian is

$$J_t^{(g)} = \frac{\partial F_\theta^{(g)}(\boldsymbol{s}_{t-1}^{(g)}, \boldsymbol{v}_t^{(g)})}{\partial \boldsymbol{s}_{t-1}^{(g)}}. \tag{19}$$

For a unit tangent vector $\boldsymbol{q}_{t-1}^{(g)}$, we compute

$$\widetilde{\boldsymbol{q}}_t^{(g)} = J_t^{(g)} \boldsymbol{q}_{t-1}^{(g)}, \qquad r_t^{(g)} = \sqrt{\left\|\widetilde{\boldsymbol{q}}_t^{(g)}\right\|_2^2 + \epsilon}, \qquad \boldsymbol{q}_t^{(g)} = \frac{\widetilde{\boldsymbol{q}}_t^{(g)}}{r_t^{(g)}}. \tag{20}$$

We implement this Jacobian–vector product using the closed-form updates provided in section C, without materializing a full Jacobian. After discarding a transient of T_0 steps, the group-wise finite-time Lyapunov estimate is

$$\widehat{\lambda}^{(g)}(\boldsymbol{x}) = \frac{1}{T - T_0} \sum_{t=T_0+1}^{T} \log(r_t^{(g)} + \epsilon). \tag{21}$$

The resulting signature is

$$\boldsymbol{\Lambda}(\boldsymbol{x}) = [\widehat{\lambda}^{(1)}(\boldsymbol{x}), \ldots, \widehat{\lambda}^{(G)}(\boldsymbol{x})]. \tag{22}$$

The block-diagonal recurrent Jacobian makes $\boldsymbol{\Lambda}(\boldsymbol{x})$ a structure-aligned directional stability descriptor; Appendix A provides its derivation and perturbation bound.

3.5 Lyapunov-Regularized Training and Inference

For a clean sample $\boldsymbol{x}$ and its nuisance-preserving augmentation $\widetilde{\boldsymbol{x}}$, we encourage stability invariance through

$$\mathcal{L}_{\text{inv}} = \mathbb{E}\left[\|\boldsymbol{\Lambda}(\boldsymbol{x}) - \boldsymbol{\Lambda}(\widetilde{\boldsymbol{x}})\|_1\right]. \tag{23}$$

To prevent signature collapse, different-label samples are separated:

$$\mathcal{L}_{\text{sep}} = \mathbb{E}_{i<j,\, y_i \neq y_j}\left[m_\lambda - \|\boldsymbol{\Lambda}(\boldsymbol{x}_i) - \boldsymbol{\Lambda}(\boldsymbol{x}_j)\|_2\right]_+. \tag{24}$$

We further discourage extreme finite-time contraction or expansion:

$$\mathcal{L}_{\text{band}} = \frac{1}{G}\sum_{g=1}^{G}\left(\left[\widehat{\lambda}^{(g)} - \lambda_{\max}\right]_+^2 + \left[\lambda_{\min} - \widehat{\lambda}^{(g)}\right]_+^2\right), \tag{25}$$

averaged over the clean and augmented views.

The classification and total objectives are

$$\mathcal{L}_{\text{cls}} = \tfrac{1}{2}\,\text{CE}(f(\boldsymbol{x}), y) + \tfrac{1}{2}\,\text{CE}(f(\widetilde{\boldsymbol{x}}), y), \tag{26}$$

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \omega_e\left(\gamma_{\text{inv}}\mathcal{L}_{\text{inv}} + \gamma_{\text{sep}}\mathcal{L}_{\text{sep}} + \gamma_{\text{band}}\mathcal{L}_{\text{band}}\right), \tag{27}$$

where ω_e gradually introduces the Lyapunov objectives. Exact coefficients and schedules are given in APPENDIX C.

The classifier consumes $\boldsymbol{z}$ rather than $\boldsymbol{\Lambda}$. Therefore, tangent propagation and all FTLE-related objectives are removed at inference. Lyapunov geometry serves only as a training-time inductive bias and introduces no FTLE-specific test-time computation.

Table 1: Comparison with state-of-the-art methods on event-based object classification. Accuracy (%) is reported. Our results are mean ± std over three seeds (2024, 2025, 2026). ♣, ♦, and ♡ denote spike-based methods, voxel/graph-based methods, and frame-based methods. Best and second-best results are marked in **red** and **blue**. "–" denotes unreported results.

Method	Venue	N-MNIST	N-Caltech101	N-CARS	ASL-DVS
♣ NDA (Li et al., 2022)	ECCV	–	78.2	90.1	–
♣ VPT-STS (Shen et al., 2023)	ICASSP	–	79.2	95.8	–
♣ EventRPG (Sun et al., 2024)	ICLR	–	**85.5**	96.0	–
♣ f-SNN (Ge et al., 2026)	ICLR	–	76.3	99.2	–
♦ GIN (Xu et al., 2019)	ICLR	75.4	47.6	84.6	51.4
♦ EventNet (Sekikawa et al., 2019)	CVPR	75.2	42.5	75.9	83.3
♦ RG-CNNs (Bi et al., 2020)	TIP	99.0	65.7	91.4	90.1
♦ EV-VGCNN (Deng et al., 2022b)	CVPR	99.4	74.8	95.3	98.3
♦ VMV-GCN (Xie et al., 2022)	RA-L	99.5	77.8	93.2	98.9
♦ TORE (Baldwin et al., 2023)	TPAMI	99.4	79.8	94.5	**99.9**
♦ SECNet (Ren et al., 2026)	ICML	**99.7**	82.4	94.7	**99.9**
♡ HATS (Sironi et al., 2018)	CVPR	99.1	64.2	90.2	–
♡ EST (Gehrig et al., 2019)	ICCV	99.0	75.3	91.9	97.9
♡ AMAE (Deng et al., 2020)	RA-L	98.3	69.4	93.6	98.4
♡ M-LSTM (Cannici et al., 2020)	ECCV	98.6	73.8	92.7	98.0
♡ MVF-Net (Deng et al., 2022a)	TCSVT	98.1	68.7	92.7	97.1
♡ EvT (Sabater et al., 2022)	CVPR	98.3	61.3	89.6	**99.9**
♡ DVS-ViT (Wang et al., 2022)	ICIP	98.1	63.3	90.7	96.9
♡ EventFormer (Kamal et al., 2023)	ICLR	94.3	84.8	–	–
♡ TOKEN (Jiang et al., 2024)	ICASSP	**99.9**	81.6	95.4	**99.9**
♡ ACDF (Chen et al., 2025)	ICML	97.4	84.4	**99.9**	**99.2**
LRDR (Ours)	–	**99.7±0.02**	**85.7±0.03**	**99.3 ± 0.04**	**99.9±0.02**

4 Experiments

4.1 Experimental Setup

We evaluate LRDR on the N-MNIST, N-Caltech101, N-CARS, and ASL-DVS datasets, using a fixed stratified 3:1:1 split for training, validation, and test sets, with a split random seed of 2026. Unless otherwise stated, LRDR and all architecture-matched variants use $T = 10$ temporal bins, 128×128 inputs, an ImageNet-pretrained ResNet-34, a projection dimension of $D = 128$, $G = 8$ dynamical groups, a transient length of $T_0 = 2$, and classifier dropout of 0.4. We train for at most 15 epochs using AdamW with batch size 12, backbone and newly initialized layer learning rates of 5×10^{-5} and 3×10^{-4}, respectively, weight decay 10^{-4}, cosine learning-rate decay, label smoothing of 0.05, mixed precision, and gradient clipping at 5.0. The backbone is frozen during the first epoch. Lyapunov regularization is disabled for the first two epochs, assigned half weight at epoch 3, and fully activated from epoch 4 onward. Early stopping monitors validation accuracy, using validation loss as a tie breaker, with a patience of five epochs.

Training uses mild event-level augmentations consisting of event dropout up to 7.5%, polarity flips up to 1.5%, timestamp jitter up to 187.5 μs, and temporal-rate perturbation around unity. These augmentations are substantially weaker than the corruption severities used for test-time robustness evaluation, which are applied only at test time. The complete model configuration, Lyapunov coefficients, and augmentation ranges are provided in APPENDIX C.

4.2 Classification Results

As shown in Table 1, LRDR delivers strong and consistent performance across all four benchmarks, with standard deviations of only 0.02–0.04 over three training seeds. On N-Caltech101, it establishes the highest reported accuracy of 85.7%, outperforming EventRPG, EventFormer, and ACDF by 0.2, 0.9, and 1.3 percentage points, respectively. This new best result highlights the effectiveness of LRDR on a challenging dataset with substantial intra-class variation.

Table 2: Model complexity of representative event-based object classification methods. Average inference time is reported on N-CARS.

Method	Params (M)	MACs (G)	Time (ms)
♦ PointNet++ (Qi et al., 2017)	1.8	4.0	211.7
♦ RG-CNNs (Bi et al., 2020)	19.5	0.8	–
♦ EV-VGCNN (Deng et al., 2022b)	0.8	0.7	14.9
♦ VMV-GCN (Xie et al., 2022)	0.8	1.3	13.2
♡ EST (Gehrig et al., 2019)	21.4	4.3	13.4
♡ M-LSTM (Cannici et al., 2020)	21.4	4.3	13.7
♡ MVF-Net (Deng et al., 2022a)	33.6	5.6	21.2
♡ ACDF (Chen et al., 2025)	21.9	3.7	4.4
LRDR (Ours)	**21.5**	**1.2**	**8.5**

Table 3: Unified robustness comparison on N-Caltech101. External baselines use the same corruption protocol, while LRDR variants additionally share the same backbone, preprocessing, split, and training protocol. Accuracy (%) is reported. mCA averages all corruption types and severity levels. Drop is the relative degradation, $100 \times (\mathrm{Clean} - \mathrm{mCA})/\mathrm{Clean}$; lower is better. Best results are marked in **red**.

Method	Venue	Clean	Dropout	Polarity	Jitter	Rate	mCA	Drop↓
External baselines								
EST (Gehrig et al., 2019)	ICCV	75.33	67.14	73.89	74.41	73.18	72.15	4.22%
EventFormer (Kamal et al., 2023)	ICLR	84.81	76.93	77.01	78.91	75.52	77.09	9.10%
ACDF (Chen et al., 2025)	ICML	84.42	77.08	82.57	82.90	82.15	81.17	3.84%
SECNet (Ren et al., 2026)	ICML	82.46	75.24	80.02	81.17	78.95	78.84	4.39%
Architecture-matched LRDR variants								
Temporal Avg	–	83.46	77.63	83.03	83.69	83.30	81.91	1.86%
Dynamics	–	83.40	77.13	82.91	83.51	81.73	81.32	2.49%
Aug-only	–	85.12	79.95	84.75	85.25	84.37	83.58	**1.81%**
Feature Inv	–	85.29	80.24	84.69	85.37	84.32	83.65	1.92%
LRDR	–	**85.74**	**80.48**	**85.20**	**85.69**	**84.46**	**83.96**	2.08%

LRDR also ranks among the leading methods on the other benchmarks. On N-CARS, it achieves 99.3%, outperforming f-SNN, TOKEN, and TORE while remaining within 0.6 percentage points of the best result. On N-MNIST, LRDR reaches 99.7%, matching SECNet and approaching the best result within 0.2 percentage points. On ASL-DVS, it achieves 99.9% and matches the highest reported accuracy. Overall, LRDR performs at or near the state of the art across event-based digit, object, vehicle, and gesture recognition.

4.3 Complexity Analysis

Table 2 compares the model complexity of representative event-based object classification methods. Following prior work, we report the number of trainable parameters, multiply–accumulate operations (MACs), and average inference time on N-CARS, measured on an NVIDIA Quadro RTX 6000. MACs are profiled over the complete $T = 10$-bin input: all bins are jointly aggregated for a single ResNet-34 forward pass, while the ordered sequence is processed by the lightweight temporal branch. Thus, the reported 1.2G MACs correspond to one complete event sample rather than one temporal bin. LRDR contains 21.5M trainable parameters and, compared with ACDF, reduces the computational cost from 3.7G to 1.2G MACs with a comparable parameter count. FTLE/JVP computation is used only during training and introduces no inference-time overhead.

LRDR takes 8.5 ms to recognize one event sample, corresponding to a throughput of approximately 118 samples per second. Although its measured latency is higher than that of several purely frame-based pipelines, LRDR maintains moderate runtime with substantially fewer MACs. This discrepancy reflects that MAC count alone does not fully capture dynamical event processing, where temporal aggregation and state updates introduce additional memory and kernel overhead. Overall, LRDR

(a) Variant configuration

Variant	Dyn.	Aug.	Feature Inv.	FTLE Inv.	Sep.	Band
Temporal Avg						
Dynamics	●					
Aug-only	●	●				
Feature Inv	●	●	●			
Inv	●	●		●		
Sep-only	●	●			●	
Band-only	●	●				●
Inv+Sep	●	●		●	●	
Full LRDR	●	●		●	●	●

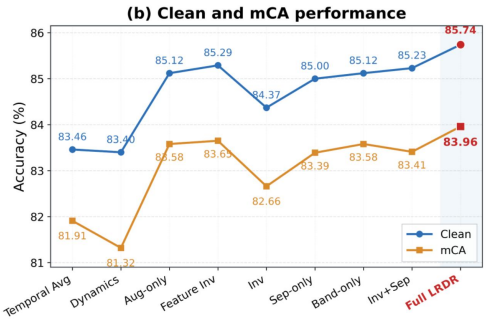


Figure 2: Architecture-matched ablation study on N-Caltech101. (a) Configurations of the evaluated variants. (b) Clean accuracy and mean corruption accuracy (mCA). Full LRDR achieves the best performance under both metrics.

achieves a favorable trade-off between classification accuracy, model complexity, and practical inference efficiency.

4.4 Robustness under Event Corruptions

We evaluate test-time robustness to event dropout $q \in \{0.1, 0.3, 0.5, 0.7\}$, polarity flips $q_p \in \{0.05, 0.10, 0.20, 0.30\}$, timestamp jitter $\sigma_t \in \{0.25, 0.5, 1, 2\}$ ms, and temporal-rate shifts $\alpha \in \{0.5, 0.75, 1.25, 1.5\}$. We average accuracy over four severity levels per family, and mCA averages the family scores. Drop is $100(\text{Clean} - \text{mCA})/\text{Clean}$. We compare LRDR with EST, EventFormer, ACDF, and SECNet under identical corruptions. To connect the external comparison with controlled ablations, we also evaluate four architecture-matched variants: *Temporal Avg* removes nonlinear dynamics, *Dynamics* retains grouped dynamics without paired augmentation or Lyapunov regularization, *Aug-only* introduces paired event augmentation without phase regularization, and *Feature Inv* additionally enforces invariance on the normalized representations $\boldsymbol{z}$. LRDR further regularizes group-wise FTLE signatures through the complete invariance, separation, and stability objectives. All internal variants share the same backbone, preprocessing, split, and optimization protocol.

As shown in Table 3, LRDR achieves the highest clean and all corruption-family accuracies, with 85.74% clean accuracy and 83.96% mCA. It surpasses ACDF by 1.79 mCA points and reduces relative Drop from 3.84% to 2.08%. LRDR also improves mCA over *Dynamics*, *Feature Inv*, and *Aug-only* by 2.64, 0.31, and 0.38 points, respectively. Although *Aug-only* has a slightly lower Drop, its absolute accuracies remain lower. The weaker performance of *Dynamics* further suggests that unconstrained temporal evolution may introduce excessive flexibility and destabilize the learned representations, motivating the proposed FTLE regularization.

4.5 Ablations: What Actually Causes the Gain?

We perform architecture-matched ablations on N-Caltech101 using the same ResNet-34 backbone, event preprocessing, temporal fusion, optimization recipe, and fixed data split. The definitions of *Temporal Avg*, *Dynamics*, *Aug-only*, and *Feature Inv* follow the robustness analysis in Table 3. We further isolate the Lyapunov objectives: *Inv*, *Sep*, and *Band* retain only the corresponding FTLE objective; *Inv+Sep* combines invariance and separation without the stability-band term; and *Full LRDR* applies all three objectives.

As shown in Figure 2, full LRDR achieves the best clean accuracy and mCA. Neither invariance nor separation alone improves over Aug-only, while Band-only closely matches it. However, adding the stability-band objective to Inv+Sep raises mCA from 83.41% to 83.96%. This suggests that the three Lyapunov objectives act complementarily: invariance and separation reshape the dynamical geometry, while the band constraint helps regularize the resulting dynamics.

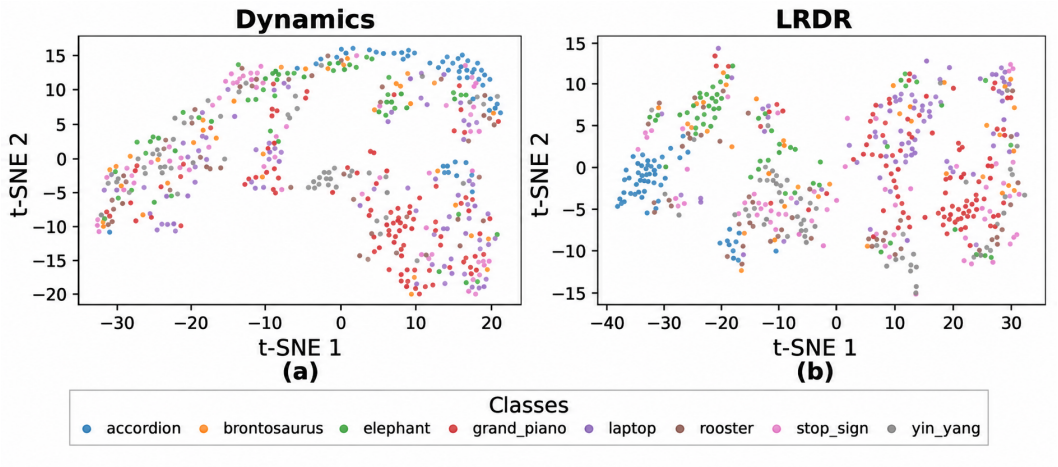


Figure 3: t-SNE visualization of learned FTLE signatures on representative N-Caltech101 classes. Compared with the unregularized Dynamics baseline, LRDR exhibits a more structured class-dependent organization in the FTLE space, consistent with the increased phase separation measured in the original dynamical signature space.

4.6 Dynamical Signature Analysis

We further examine the learned group-wise FTLE signature $\Lambda(x)$. Figure 3 visualizes eight representative N-Caltech101 classes using t-SNE. We quantify class-dependent organization using the intra-class and inter-class distances defined in Appendix D. On the full held-out test set, LRDR increases the phase-separation margin $M_\lambda = D_{\text{inter}} - D_{\text{intra}}$ from 0.002 to 0.086, while $R_\lambda = D_{\text{inter}}/(D_{\text{intra}} + \epsilon)$ increases from approximately 1.33 to 1.90. These results are consistent with stronger class-dependent organization of the learned dynamical signatures.

4.7 Discussion

LRDR combines pretrained semantic features with learned temporal dynamics. The grouped recurrence models the temporal evolution of event features, while the residual update in eq. (12) preserves useful semantic information. Appendix A links FTLE regularization to bounded perturbation amplification, no FTLE computation is used at inference. Comparisons with augmentation and feature invariance further support the role of temporal stability in improving robustness.

The signature in eq. (22) is a group-wise finite-time stability descriptor designed to capture sample-conditioned directional growth within the learned recurrent dynamics. Its estimation depends on the temporal horizon, group structure, and tangent direction, and incurs additional training cost. Moreover, synthetic corruptions cannot fully represent real sensor shifts, motivating future evaluation on broader event tasks and natural distribution shifts.

5 Conclusion

This paper proposes a novel event processing method centered on our Lyapunov-Regularized Dynamical Representation Learning framework, LRDR, combined with CCNN to model the temporal evolution of event features. LRDR uses learnable nonlinear dynamics and group-wise FTLEs to explicitly characterize and constrain dynamical stability. During training, perturbation consistency, inter-class separation, and bounded stability are jointly optimized to improve robustness and discriminability. CCNN captures the nonlinear state evolution of event features and provides dynamical trajectories for FTLE-based stability analysis. Experiments on multiple event recognition benchmarks and under various perturbations show that LRDR achieves SOTA classification performance and good robustness. Overall, finite-time stability serves as an effective training constraint for learning stable and robust event representations.

Reproducibility Statement

The model architecture, recurrent dynamics, finite-time Lyapunov exponent computation, and training objectives are fully specified in Sections 3 and Appendix A. Section 4 provides the dataset splits, training protocol, evaluation metrics, and corruption settings. Appendix C further reports the model configuration, optimization hyperparameters, and the exact tangent propagation used for FTLE estimation. The main benchmark results reported in Table 1 are averaged over three training seeds.

AI Use Statement

Generative AI tools were used to assist with language polishing of the manuscript. All text generated with AI assistance was independently checked by the authors. The authors take full responsibility for the final contents of the submission.

References

R. Wes Baldwin, Ruixu Liu, Mohammed Almatrafi, Vijayan Asari, and Keigo Hirakawa. Time-ordered recent event (tore) volumes for event cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2519–2532, 2023.

Yin Bi, Aaron Chadha, Alhabib Abbas, Eirina Bourtsoulatze, and Yiannis Andreopoulos. Graph-based spatio-temporal feature learning for neuromorphic vision sensing. *IEEE Transactions on Image Processing*, 29:9084–9098, 2020.

Marco Cannici, Marco Ciccone, Andrea Romanoni, and Matteo Matteucci. A differentiable recurrent surface for asynchronous event-based data. In *European Conference on Computer Vision*, pp. 136–152. Springer, 2020.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, pp. 6571–6583, 2018.

Yu Chen, Jing Lian, Zhaofei Yu, Jizhao Liu, Jisheng Dang, and Gang Wang. A chaotic dynamics framework inspired by dorsal stream for event signal processing. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, 2025.

Yongjian Deng, Youfu Li, and Hao Chen. Amae: Adaptive motion-agnostic encoder for event-based object classification. *IEEE Robotics and Automation Letters*, 5(3):4596–4603, 2020.

Yongjian Deng, Hao Chen, and Youfu Li. Mvf-net: A multi-view fusion network for event-based object classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(12): 8275–8284, 2022a.

Yongjian Deng, Hao Chen, Hai Liu, and Youfu Li. A voxel graph cnn for object classification with event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1172–1181, 2022b.

Guillermo Gallego, Tobi Delbruck, Garrick Orchard, Chiara Bartolozzi, Ben Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jorg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180, 2022.

Chengjie Ge, Yufeng Peng, Zihao Li, Qiyu Kang, Xueyang Fu, Xuhao Li, Qixin Zhang, Junhao Ren, and Zheng-Jun Zha. Fractional-order spiking neural network. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=NJhBSLJ0nL.

Daniel Gehrig, Antonio Loquercio, Konstantinos G. Derpanis, and Davide Scaramuzza. End-to-end learning of representations for asynchronous event-based data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5633–5643, 2019.

Jesse Hagenaars, Federico Paredes-Valles, and Guido C. H. E. de Croon. Self-supervised learning of event-based optical flow with spiking neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 7167–7179, 2021.

Bin Jiang, Zhihao Li, M. Salman Asif, Xun Cao, and Zhan Ma. Token-based spatiotemporal representation of the events. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5240–5244, 2024.

Uday Kamal, Saurabh Dash, and Saibal Mukhopadhyay. Associative memory augmented asynchronous spatiotemporal representation learning for event-based perception. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ZCStthyW-TD.

Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Xavier Lagorce, Garrick Orchard, Francesco Galluppi, Bertram E. Shi, and Ryad B. Benosman. Hots: A hierarchy of event-based time-surfaces for pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7):1346–1359, 2017.

Yuhang Li, Youngeun Kim, Hyoungseob Park, Tamar Geller, and Priyadarshini Panda. Neuromorphic data augmentation for training spiking neural networks. In *European Conference on Computer Vision*, pp. 631–649, 2022.

Jizhao Liu, Jing Lian, Julien C. Sprott, Qidong Liu, and Yide Ma. The butterfly effect in primary visual cortex. *IEEE Transactions on Computers*, 71(11):2803–2815, 2022.

Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5099–5108, 2017.

Hongwei Ren, Fei Ma, Xiaopeng Lin, Yuetong Fang, Hongxiang Huang, Yue Zhou, Yulong Huang, Haotian Fu, Ziyi Yang, Youxin Jiang, Xiangqian Wu, and Bojun Cheng. Scalable event cloud network for event-based classification. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Alberto Sabater, Luis Montesano, and Ana C. Murillo. Event transformer: A sparse-aware solution for efficient event data processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2677–2686, 2022.

Samuel S. Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation. In *International Conference on Learning Representations*, 2017.

Yusuke Sekikawa, Kosuke Hara, and Hideo Saito. Eventnet: Asynchronous recursive event processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3887–3896, 2019.

Haibo Shen, Juyu Xiao, Yihao Luo, Xiang Cao, Liangqi Zhang, and Tianjiang Wang. Training robust spiking neural networks with viewpoint transform and spatiotemporal stretching. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1–5, 2023.

Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. Hats: Histograms of averaged time surfaces for robust event-based object classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1731–1740, 2018.

Mingyuan Sun, Donghao Zhang, Zongyuan Ge, Jiaxu Wang, Jia Li, Zheng Fang, and Renjing Xu. EventRPG: Event data augmentation with relevance propagation guidance. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=i7LCsDMcZ4.

Zuowen Wang, Yuhuang Hu, and Shih-Chii Liu. Exploiting spatial sparsity for event cameras with visual transformers. In *IEEE International Conference on Image Processing*, pp. 411–415, 2022.

Tobias Wöhrer and Christian Kuehn. Tracking Finite-Time Lyapunov Exponents to robustify Neural ODEs. *arXiv preprint arXiv:2602.09613*, 2026. doi: 10.48550/arXiv.2602.09613. URL https://arxiv.org/abs/2602.09613.

Bochen Xie, Yongjian Deng, Zhaoliang Shao, Hai Liu, and Youfu Li. Vmv-gcn: Volumetric multi-view based graph cnn for event stream classification. *IEEE Robotics and Automation Letters*, 7(2): 1976–1983, 2022.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

A Structural Interpretation of Group-Wise Finite-Time Stability

This section provides a structural interpretation of the group-wise finite-time Lyapunov signature. We first show that the grouping follows the block structure of the recurrent-state Jacobian, then establish the exact directional-growth quantity measured by $\widehat{\lambda}^{(g)}$, and finally relate finite-time expansion to the propagation of event-induced feature perturbations.

A.1 Structure-Aligned Group-Wise Decomposition

For group g, define

$$A_f^{(g)} = \operatorname{diag}\left(\boldsymbol{a}_f^{(g)}\right), \qquad A_e^{(g)} = \operatorname{diag}\left(\boldsymbol{a}_e^{(g)}\right), \tag{28}$$

$$V_e^{(g)} = \operatorname{diag}\left(\boldsymbol{v}_e^{(g)}\right), \qquad D_t^{(g)} = \operatorname{diag}\left(\boldsymbol{\beta}^{(g)} \odot \boldsymbol{y}_t^{(g)} \odot (1 - \boldsymbol{y}_t^{(g)})\right). \tag{29}$$

The state Jacobian associated with $\boldsymbol{s}_t^{(g)} = [\boldsymbol{u}_t^{(g)}; \boldsymbol{e}_t^{(g)}; \boldsymbol{y}_t^{(g)}]$ is

$$J_t^{(g)} = \begin{bmatrix} A_f^{(g)} & 0 & 0 \\ 0 & A_e^{(g)} & V_e^{(g)} \\ D_t^{(g)} A_f^{(g)} & -D_t^{(g)} A_e^{(g)} & -D_t^{(g)} V_e^{(g)} \end{bmatrix}. \tag{30}$$

Although the parameter matrices in Eq. (30) are shared across samples, $D_t^{(g)}$ depends on the input-conditioned response $\boldsymbol{y}_t^{(g)}(\boldsymbol{x})$. Hence, the resulting Jacobian and its finite-time growth are sample dependent.

Conditioned on the encoded input sequence, each recurrent group depends only on its own previous state. Therefore,

$$\frac{\partial \boldsymbol{s}_t^{(g)}}{\partial \boldsymbol{s}_{t-1}^{(g')}} = 0, \qquad g \neq g', \tag{31}$$

and the complete recurrent-state Jacobian is block diagonal:

$$J_t = \operatorname{blkdiag}\left(J_t^{(1)}, \ldots, J_t^{(G)}\right). \tag{32}$$

Consequently, the finite-time transition matrix satisfies

$$\Phi_{T,k} = J_T J_{T-1} \cdots J_{k+1} \tag{33}$$

$$= \operatorname{blkdiag}\left(\Phi_{T,k}^{(1)}, \ldots, \Phi_{T,k}^{(G)}\right), \tag{34}$$

where $\Phi_{T,k}^{(g)} = J_T^{(g)} \cdots J_{k+1}^{(g)}$. Its spectral norm obeys

$$\|\Phi_{T,k}\|_2 = \max_g \|\Phi_{T,k}^{(g)}\|_2. \tag{35}$$

Thus, the group-wise signature follows the intrinsic block structure of the recurrent dynamics rather than an arbitrary decomposition of a single global exponent. This decomposition applies to the recurrent state Jacobian conditioned on the encoded inputs, not to the Jacobian of the complete dual-branch network.

A.2 Directional Finite-Time Growth Identity

Consider the tangent propagation after the transient period:

$$\widetilde{\boldsymbol{q}}_t^{(g)} = J_t^{(g)} \boldsymbol{q}_{t-1}^{(g)}, \tag{36}$$

$$r_t^{(g)} = \left\| \widetilde{\boldsymbol{q}}_t^{(g)} \right\|_2, \tag{37}$$

$$\boldsymbol{q}_t^{(g)} = \frac{\widetilde{\boldsymbol{q}}_t^{(g)}}{r_t^{(g)}}. \tag{38}$$

Here we omit the numerical stabilizer ϵ to state the exact identity; the implementation adds ϵ only to avoid numerical degeneracy. Repeated substitution gives

$$\Phi_{T,T_0}^{(g)} \boldsymbol{q}_{T_0}^{(g)} = \left(\prod_{t=T_0+1}^{T} r_t^{(g)} \right) \boldsymbol{q}_T^{(g)}. \tag{39}$$

Because $\|\boldsymbol{q}_T^{(g)}\|_2 = 1$,

$$\prod_{t=T_0+1}^{T} r_t^{(g)} = \left\| \Phi_{T,T_0}^{(g)} \boldsymbol{q}_{T_0}^{(g)} \right\|_2. \tag{40}$$

Therefore,

$$\widehat{\lambda}^{(g)}(\boldsymbol{x}) = \frac{1}{T - T_0} \sum_{t=T_0+1}^{T} \log r_t^{(g)} \tag{41}$$

$$= \frac{1}{T - T_0} \log \left\| \Phi_{T,T_0}^{(g)}(\boldsymbol{x}) \boldsymbol{q}_{T_0}^{(g)} \right\|_2. \tag{42}$$

Hence, $\widehat{\lambda}^{(g)}$ is exactly the finite-time logarithmic growth rate along the propagated tangent direction of group g. It is related to the maximal group-wise FTLE by

$$\widehat{\lambda}^{(g)}(\boldsymbol{x}) \leq \lambda_{\max}^{(g)}(\boldsymbol{x}) = \frac{1}{T - T_0} \log \left\| \Phi_{T,T_0}^{(g)}(\boldsymbol{x}) \right\|_2. \tag{43}$$

The proposed signature should therefore be interpreted as a collection of sample-conditioned directional growth rates, rather than a complete Lyapunov spectrum.

A.3 Propagation of Event-Induced Perturbations

Let $\boldsymbol{x}$ and $\widetilde{\boldsymbol{x}}$ denote a clean event sample and a perturbed view, and define

$$\delta \boldsymbol{s}_t^{(g)} = \boldsymbol{s}_t^{(g)}(\widetilde{\boldsymbol{x}}) - \boldsymbol{s}_t^{(g)}(\boldsymbol{x}), \tag{44}$$

$$\delta \boldsymbol{v}_t^{(g)} = \boldsymbol{v}_t^{(g)}(\widetilde{\boldsymbol{x}}) - \boldsymbol{v}_t^{(g)}(\boldsymbol{x}). \tag{45}$$

A first-order expansion of the driven state transition gives

$$\delta \boldsymbol{s}_t^{(g)} = J_t^{(g)} \delta \boldsymbol{s}_{t-1}^{(g)} + B_t^{(g)} \delta \boldsymbol{v}_t^{(g)} + \mathcal{O}(\|\delta\|^2), \tag{46}$$

where

$$B_t^{(g)} = \frac{\partial F_\theta^{(g)}(\boldsymbol{s}_{t-1}^{(g)}, \boldsymbol{v}_t^{(g)})}{\partial \boldsymbol{v}_t^{(g)}}. \tag{47}$$

The clean and perturbed trajectories use the same zero initial state, so unrolling Eq. (46) yields

$$\delta \boldsymbol{s}_T^{(g)} = \sum_{k=1}^{T} \Phi_{T,k}^{(g)} B_k^{(g)} \delta \boldsymbol{v}_k^{(g)} + \mathcal{O}(\|\delta\|^2), \tag{48}$$

with $\Phi_{T,T}^{(g)} = I$. For $k < T$, define the maximal finite-time expansion rate

$$\lambda_{T,k}^{(g),\max} = \frac{1}{T - k} \log \left\| \Phi_{T,k}^{(g)} \right\|_2. \tag{49}$$

Taking norms in Eq. (48) gives

$$\left\|\delta \boldsymbol{s}_T^{(g)}\right\|_2 \lesssim \left\|B_T^{(g)}\right\|_2 \left\|\delta \boldsymbol{v}_T^{(g)}\right\|_2 \tag{50}$$

$$+ \sum_{k=1}^{T-1} \exp\Big((T-k)\lambda_{T,k}^{(g),\max}\Big) \left\|B_k^{(g)}\right\|_2 \left\|\delta \boldsymbol{v}_k^{(g)}\right\|_2 . \tag{51}$$

Equation (51) shows that finite-time expansion determines how strongly an input-induced feature disturbance is amplified by subsequent recurrent updates. Event dropout and polarity flips perturb feature content, whereas timestamp jitter and rate shifts perturb temporal organization; after temporal encoding, all appear as disturbances $\delta \boldsymbol{v}_t^{(g)}$.

The directional FTLE in Eq. (42) provides a tractable differentiable proxy for these group-wise expansion rates. Accordingly, $\mathcal{L}_{\text{band}}$ discourages extreme expansion or contraction, $\mathcal{L}_{\text{inv}}$ aligns the stability profiles of clean and augmented views, and $\mathcal{L}_{\text{sep}}$ prevents stable signatures from collapsing across classes.Together, these results show how the learned finite-time expansion profile regulates the propagation of event-induced disturbances through the recurrent dynamics. The analysis provides a local first-order characterization; extending it to global certified guarantees is left to future work.

B Rich Dynamical Properties of CCNN

B.1 Continuous-Coupled Neural Network

The continuous-coupled neural network (CCNN) is a nonlinear dynamical model inspired by the primary visual cortex. Different from PCNNs, which generate discrete spikes through a deterministic threshold mechanism, CCNN replaces the firing process with a continuous probabilistic activation, allowing the model to capture richer nonlinear temporal dynamics. This section provides a brief proof of the periodic and chaotic dynamical properties of CCNN. For the detailed proof, please refer to Liu et al. (2022)

The CCNN state evolution is described as:

$$\begin{aligned}
F_{ij}(n) &= e^{-\alpha_f} F_{ij}(n-1) + V_F M_{ijkl} Y_{kl}(n-1) + S_{ij}, \\
L_{ij}(n) &= e^{-\alpha_l} L_{ij}(n-1) + V_L W_{ijkl} Y_{kl}(n-1), \\
U_{ij}(n) &= F_{ij}(n)(1 + \beta L_{ij}(n)), \\
Y_{ij}(n) &= \frac{1}{1 + e^{-(U_{ij}(n) - E_{ij}(n))}}, \\
E_{ij}(n) &= e^{-\alpha_e} E_{ij}(n-1) + V_E Y_{ij}(n-1).
\end{aligned} \tag{52}$$

where F_{ij} denotes the feeding activity, L_{ij} represents the coupling activity, U_{ij} is the modulated internal state, E_{ij} denotes the adaptive dynamic threshold, and Y_{ij} is the continuous output response. The parameters α_f, α_l, and α_e control temporal decay, while M and W represent feeding and linking connections, respectively.

B.2 Periodic Dynamics of CCNN under Constant Stimulus

To demonstrate the intrinsic dynamical capability of CCNN, we first analyze its behavior under a constant stimulus. Following the original CCNN formulation, we consider a simplified non-linking neuron model:

$$\begin{aligned}
U(t) &= e^{-\alpha_f} U(t-1) + S, \\
Y(t) &= \frac{1}{1 + e^{-(U(t) - E(t))}}, \\
E(t) &= e^{-\alpha_e} E(t-1) + V_E Y(t-1).
\end{aligned} \tag{53}$$

where $U(t)$ denotes the internal activity, $E(t)$ is the dynamic threshold, and $Y(t)$ is the continuous output response. α_f, α_e, and V_E control the temporal evolution of the neuron states.

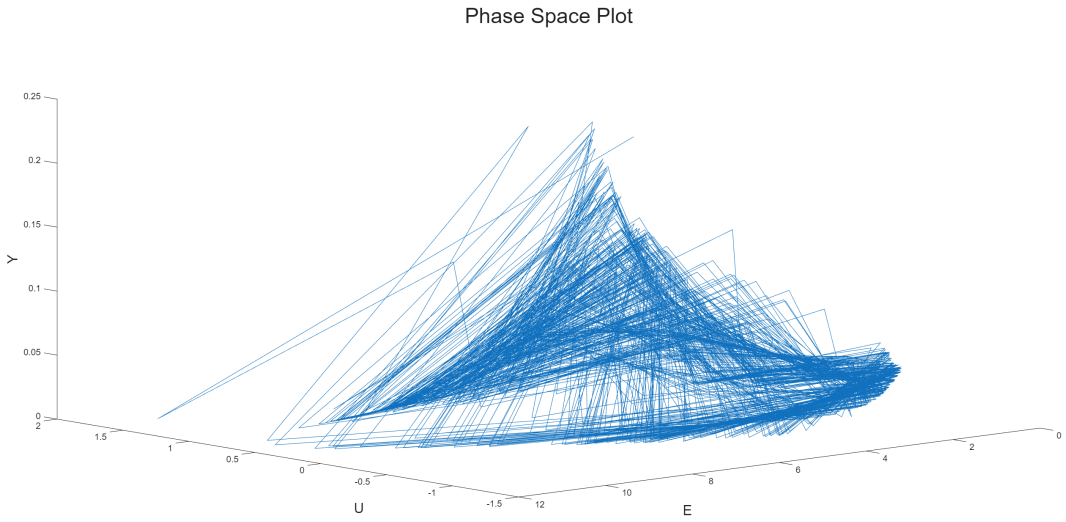


Figure 4: Phase-space trajectory of CCNN under periodic stimulus $(0.4 \sin t)$. The non-repeating attractor indicates nonlinear dynamical behavior.

For a constant external stimulus S, the evolution of $U(t)$ is independent of the output state and follows a geometric recurrence:

$$U(t) = S\frac{1 - e^{-t\alpha_f}}{1 - e^{-\alpha_f}}. \tag{54}$$

Meanwhile, the threshold variable $E(t)$ is coupled with the output response:

$$E(t) = e^{-\alpha_e}E(t-1) + \frac{V_E}{1 + e^{-(U(t-1)-E(t-1))}}. \tag{55}$$

Assuming the neuron returns to the same threshold state after T iterations,

$$E(t+T) = E(t), \tag{56}$$

the periodic condition can be obtained by substituting the recurrence relation of $E(t)$:

$$E(0) = e^{-\alpha_e}E(0) + \frac{V_E}{1 + e^{-(U(T-1)-E(0))}}. \tag{57}$$

Therefore, the corresponding activity state satisfies

$$U(T-1) = E(0) - \ln\left(\frac{V_E}{(1 - e^{-\alpha_e})E(0)} - 1\right). \tag{58}$$

Combining this equation with the analytical solution of $U(t)$ gives the oscillation period:

$$T = 1 + \frac{1}{\alpha_f}\ln\left(\frac{S}{S - (1 - e^{-\alpha_f})\left(E(0) - \ln\left(\frac{V_E}{(1-e^{-\alpha_e})E(0)} - 1\right)\right)}\right). \tag{59}$$

This analysis indicates that under a constant stimulus, the CCNN neuron converges to a periodic oscillatory state. The oscillation frequency is determined by the stimulus intensity and the intrinsic dynamic parameters. A larger stimulus amplitude produces a shorter period and a higher oscillation frequency.

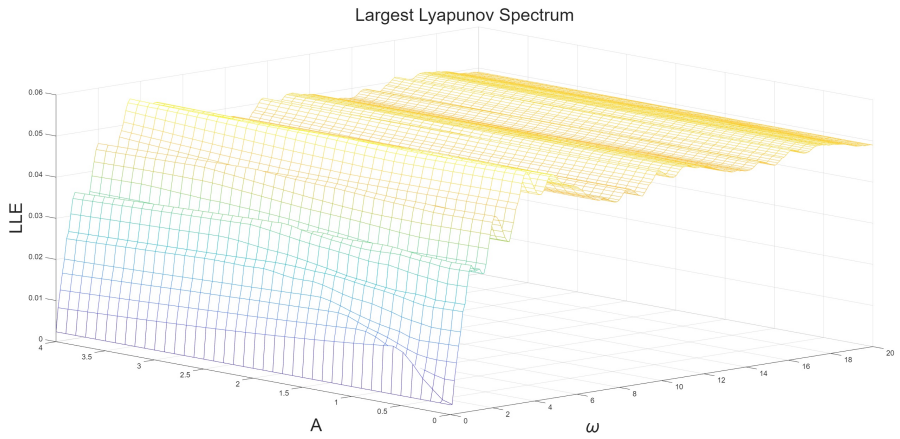


Figure 5: Largest Lyapunov exponent of CCNN under periodic stimulus $(0.4 \sin t)$. A positive LLE indicates sensitive dependence on initial conditions and chaotic dynamics.

B.3 Chaotic Dynamics of CCNN under Periodic Stimulus

We further investigate the nonlinear dynamical behavior of CCNN under periodic stimulation. Following the original CCNN analysis, we consider the simplified non-linking CCNN model:

$$\begin{aligned} U(t) &= e^{-\alpha_f} U(t-1) + S(t), \\ Y(t) &= \frac{1}{1 + e^{-(U(t)-E(t))}}. \end{aligned} \tag{60}$$

$$E(t) = e^{-\alpha_e} E(t-1) + V_E Y(t-1). \tag{61}$$

Different from the constant stimulus case, the internal activity $U(t)$ is driven by a time-varying periodic signal:

$$S(t) = A \sin(t). \tag{62}$$

Suppose all values of U, Y and E are initially zero, by iterative derivation, we obtain:

$$U(n) = e^{-a_f(n-1)} S(1) + e^{-a_f(n-2)} S(2) + \cdots + S(n) = \sum_{k=1}^{n} e^{-a_f(n-k)} S(k). \tag{63}$$

By extending the discrete sequence to a continuous time domain and letting the iteration time step approach zero, the discrete summation can be converted into the integral form of continuous function:

$$U(n) = \int_0^n e^{-a_f \tau} \cos \tau \mathrm{d}\tau \tag{64}$$

We solve Eq. 64 via integration by parts and obtain:

$$U(n) = \frac{e^{-\alpha_f n} \sin n - \alpha_f \cdot e^{-\alpha_f n} \cdot \cos n}{1 + e^{-2\alpha_f}} \tag{65}$$

$E(n)$ is affected by $Y(n)$ at each iteration, which cannot be represented by a geometric sequence. Therefore, the CCNN model has distinct dynamic characteristics under a periodic stimulus.

In this part, nonlinear dynamic analysis methods are used to explore the dynamic characteristics of the CCNN under a periodic stimulus. The equilibrium point of the CCNN model is given by:

$$
\begin{aligned}
&E(n+1) = E(n) \Rightarrow \\
&E(n) = e^{-\alpha_e} E(n) + \frac{V_E}{1 + e^{-(U(n) - E(n))}} \Rightarrow \\
&E(n)(1 + e^{-(U(n) - E(n))}) = \frac{V_E}{1 - e^{-\alpha_e}}
\end{aligned} \tag{66}
$$

Taylor expansion of e^x is used to simplify Eq. 66. Here only the first two terms of the Taylor series are considered. Eq. 66 can be simplified to:

$$
E(n)(2 + E(n) - U(n)) = \frac{V_E}{1 - e^{-\alpha_e}} \Rightarrow E(n)^2 - (U(n) - 2)E(n) - \frac{V_E}{1 - e^{-\alpha_e}} = 0 \tag{67}
$$

Eq. 61 is a quadratic equation with $\Delta = (U(n) - 2)^2 + 4\frac{V_E}{1 - e^{-\alpha_e}}$. Since $V_E > 0$, $\alpha_e > 0$, $4\frac{V_E}{1 - e^{-\alpha_e}} > 0 \Rightarrow \Delta > 0$, $E(n)$ can be represented as

$$
E(n) = \frac{U(n) - 2 \pm \sqrt{(U(n) - 2)^2 + 4\frac{V_E}{1 - e^{-\alpha_e}}}}{2}. \tag{68}
$$

The derivative of Eq. 61 is given by:

$$
f(E(n)) = 2E(n) - U(n) + 2 \tag{69}
$$

Therefore, for $E(n) = \frac{U(n) - 2 + \sqrt{(U(n) - 2)^2 + 4\frac{V_E}{1 - e^{-\alpha_e}}}}{2}$, $f(E(n)) > 0$. For $E(n) = \frac{U(n) - 2 - \sqrt{(U(n) - 2)^2 + 4\frac{V_E}{1 - e^{-\alpha_e}}}}{2}$, $f(E(n)) < 0$.

Since $U(n)$ is an independent variable, Eq. 61 is a two-dimensional discrete dynamical system. According to the above analysis, it has two equilibrium points, the first of which is unstable and given by:

$$
\begin{aligned}
U(n) &= \frac{e^{-n\alpha_f} \sin(n) - \alpha_f e^{-n\alpha_f} \cos(n)}{1 + \alpha_f^2} \\
E(n) &= \frac{U(n) - 2 + \sqrt{(U(n) - 2)^2 + 4\frac{V_E}{1 - e^{-\alpha_e}}}}{2} \\
Y(n) &= \frac{1}{1 + e^{-(U(n) - E(n))}}
\end{aligned} \tag{70}
$$

The second equilibrium point is stable and is given by:

$$
\begin{aligned}
U(n) &= \frac{e^{-n\alpha_f} \sin(n) - \alpha_f e^{-n\alpha_f} \cos(n)}{1 + \alpha_f^2} \\
E(n) &= \frac{U(n) - 2 - \sqrt{(U(n) - 2)^2 + 4\frac{V_E}{1 - e^{-\alpha_e}}}}{2} \\
Y(n) &= \frac{1}{1 + e^{-(U(n) - E(n))}}
\end{aligned} \tag{71}
$$

As shown in Figure 5, a positive LLE indicates that nearby trajectories diverge exponentially, demonstrating sensitive dependence on initial conditions and chaotic dynamics.

Using the CCNN parameters $\alpha_f = 0.1$, $\alpha_e = 1$, and $V_E = 50$, with periodic stimulus $S(t) = 0.4\sin(t)$, the generated trajectory forms a non-periodic attractor in the phase space of (U, E, Y), as shown in Figure 4. Meanwhile, the computed largest Lyapunov exponent remains positive, confirming the existence of chaotic dynamics.

C Exact FTLE Propagation and Model Configuration

For completeness, we provide the component-wise tangent propagation used to evaluate eq. (20). For

$$\boldsymbol{q}_{t-1}^{(g)} = [\boldsymbol{q}_{t-1}^{u,(g)}; \boldsymbol{q}_{t-1}^{e,(g)}; \boldsymbol{q}_{t-1}^{y,(g)}],$$

holding $\boldsymbol{v}_t^{(g)}$ fixed gives

$$\widetilde{\boldsymbol{q}}_t^{u,(g)} = \boldsymbol{a}_f^{(g)} \odot \boldsymbol{q}_{t-1}^{u,(g)}, \tag{72}$$

$$\widetilde{\boldsymbol{q}}_t^{e,(g)} = \boldsymbol{a}_e^{(g)} \odot \boldsymbol{q}_{t-1}^{e,(g)} + \boldsymbol{v}_e^{(g)} \odot \boldsymbol{q}_{t-1}^{y,(g)}, \tag{73}$$

$$\widetilde{\boldsymbol{q}}_t^{y,(g)} = \boldsymbol{y}_t^{(g)} \odot (1 - \boldsymbol{y}_t^{(g)}) \odot \boldsymbol{\beta}^{(g)} \odot \left(\widetilde{\boldsymbol{q}}_t^{u,(g)} - \widetilde{\boldsymbol{q}}_t^{e,(g)}\right). \tag{74}$$

The three components are concatenated to form $\widetilde{\boldsymbol{q}}_t^{(g)}$, after which the growth factor and renormalization in eq. (20) are applied. This computation requires only vector operations within each group and does not construct the full Jacobian matrix.

FTLE initialization and numerical details. All recurrent states are initialized to zero,

$$\boldsymbol{u}_0 = \boldsymbol{e}_0 = \boldsymbol{y}_0 = \boldsymbol{0}. \tag{75}$$

For each dynamical group, the initial tangent components $\boldsymbol{q}_0^{u,(g)}$, $\boldsymbol{q}_0^{e,(g)}$, and $\boldsymbol{q}_0^{y,(g)}$ are sampled independently from a standard Gaussian distribution using the fixed random seed 12345. They are jointly normalized as

$$\boldsymbol{q}_0^{(g)} = \frac{[\boldsymbol{q}_0^{u,(g)}; \boldsymbol{q}_0^{e,(g)}; \boldsymbol{q}_0^{y,(g)}]}{\sqrt{\left\|\boldsymbol{q}_0^{u,(g)}\right\|_2^2 + \left\|\boldsymbol{q}_0^{e,(g)}\right\|_2^2 + \left\|\boldsymbol{q}_0^{y,(g)}\right\|_2^2 + \epsilon}}, \tag{76}$$

with $\epsilon = 10^{-8}$. The resulting tangent components are stored as fixed non-trainable buffers and reused for all samples. During propagation, the same $\epsilon = 10^{-8}$ is used both in the tangent growth-factor computation and before the logarithm,

$$r_t^{(g)} = \sqrt{\left\|\widetilde{\boldsymbol{q}}_t^{u,(g)}\right\|_2^2 + \left\|\widetilde{\boldsymbol{q}}_t^{e,(g)}\right\|_2^2 + \left\|\widetilde{\boldsymbol{q}}_t^{y,(g)}\right\|_2^2 + 10^{-8}}, \tag{77}$$

$$\boldsymbol{q}_t^{(g)} = \widetilde{\boldsymbol{q}}_t^{(g)} / r_t^{(g)}. \tag{78}$$

The first T_0 recurrent updates are discarded, and FTLE accumulation uses the remaining $T - T_0$ steps.

Separation-pair construction. For $\mathcal{L}_{\text{sep}}$, we first compute the complete pairwise Euclidean-distance matrix between the FTLE signatures in the current minibatch. We then retain all unordered different-label pairs,

$$\mathcal{N} = \{(i, j) : i < j,\ y_i \neq y_j\}, \tag{79}$$

and evaluate

$$\mathcal{L}_{\text{sep}} = \frac{1}{|\mathcal{N}|} \sum_{(i,j)\in\mathcal{N}} \left[m_\lambda - \|\boldsymbol{\Lambda}(\boldsymbol{x}_i) - \boldsymbol{\Lambda}(\boldsymbol{x}_j)\|_2\right]_+. \tag{80}$$

No negative-pair subsampling is used. If a minibatch contains no different-label pair, this term is set to zero.

Model and optimization configuration. The default configuration uses $T = 10$ temporal bins, 128×128 inputs, an ImageNet-pretrained ResNet-34, $D = 128$ projected features, $G = 8$ groups, $T_0 = 2$, and classifier dropout 0.4. We train for at most 15 epochs using AdamW, batch size 12, weight decay 10^{-4}, label smoothing 0.05, gradient clipping at 5.0, and cosine learning-rate decay. The backbone and newly initialized layers use learning rates of 5×10^{-5} and 3×10^{-4}, respectively.

The backbone is frozen during the first epoch. We set $\omega_e = 0$ for the first two epochs, $\omega_e = 0.5$ at epoch 3, and $\omega_e = 1$ from epoch 4 onward. The Lyapunov coefficients are

$$\gamma_{\text{inv}} = 0.03, \qquad \gamma_{\text{sep}} = 0.02, \qquad \gamma_{\text{band}} = 0.002, \tag{81}$$

with $m_\lambda = 0.10$ and $[\lambda_{\min}, \lambda_{\max}] = [-0.5, 0.5]$.

Training augmentation ranges. Training uses paired event-level augmentation before temporal binning. For the default configuration, the augmentation strength is fixed to $s = 0.75$. The event-dropout probability is sampled as

$$q \sim \mathcal{U}(0, 0.10s) = \mathcal{U}(0, 0.075), \tag{82}$$

and the polarity-flip probability is

$$q_p \sim \mathcal{U}(0, 0.02s) = \mathcal{U}(0, 0.015). \tag{83}$$

The timestamp-jitter scale is sampled as

$$\sigma_t \sim \mathcal{U}(0, 250s)\ \mu\text{s} = \mathcal{U}(0, 187.5)\ \mu\text{s}, \tag{84}$$

and the temporal-rate factor is sampled as

$$\alpha \sim \mathcal{U}\left(\max(0.7, 1 - 0.05s), 1 + 0.05s\right) = \mathcal{U}(0.9625, 1.0375). \tag{85}$$

These perturbations are substantially milder than those used in the test-time corruption benchmark.

D Dynamical Phase-Separation Metrics

To quantify class-dependent organization in the original FTLE-signature space, we define the average intra-class and inter-class distances as

$$D_{\text{intra}} = \mathbb{E}_{i<j:\, y_i = y_j} \left[\|\Lambda(x_i) - \Lambda(x_j)\|_2 \right], \tag{86}$$

$$D_{\text{inter}} = \mathbb{E}_{i<j:\, y_i \neq y_j} \left[\|\Lambda(x_i) - \Lambda(x_j)\|_2 \right]. \tag{87}$$

We then define the phase-separation margin and distance ratio as

$$M_\lambda = D_{\text{inter}} - D_{\text{intra}}, \tag{88}$$

$$R_\lambda = \frac{D_{\text{inter}}}{D_{\text{intra}} + \epsilon}, \tag{89}$$

where $\epsilon = 10^{-8}$ is used for numerical stability. Larger M_λ and R_λ indicate stronger class-dependent separation in the original FTLE-signature space.