A Unified Continuous-Time Spiking Transformer Framework for Stable and Energy-Efficient Cross-Day Neural Decoding

Ziqiang Ye[1], Erjun Xiao[1], Yu Song[1], Liyuan Han[1,2,3], Boshi Zhao[1], Tielin Zhang[1,2,3,4,5]*

1. Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences

2. State Key Laboratory of Brain Cognition and Brain-inspired Intelligence Technology

3. Institute of Neuroscience, Chinese Academy of Sciences

4. School of Artificial Intelligence, University of Chinese Academy of Sciences

5. The Key Laboratory of Cognition and Decision Intelligence for Complex Systems, Institute of Automation, Chinese Academy of Sciences

E-mail: zhangtielin@ion.ac.cn

Abstract: Cross-day variability in neural signals remains a major obstacle to achieving long-term stability in brain–computer interface (BCI) systems. To address this challenge, we propose a unified cross-day neural decoding framework that integrates direct spiking encoding, Spiking Transformer–based spatiotemporal modeling, and ODE-LSTM–based continuous-time decoding. The proposed method directly converts neural signals into spike sequences without requiring an additional encoder, effectively captures cross-channel and cross-temporal dependencies through event-driven sparse self-attention, and substantially mitigates long-timescale neural drift via continuous-time dynamical modeling. We conduct systematic evaluations on two intracortical BCI datasets—an isometric wrist force control task and a center-out reaching task. The results demonstrate that the proposed framework achieves significantly higher cross-day decoding accuracy and superior energy efficiency compared with existing deep learning models. Furthermore, the lightweight ODE-LSTM obtained through pruning and quantization preserves decoding performance while achieving an order-of-magnitude reduction in energy consumption, confirming the redundancy and stability of the learned continuous-time neural representations and highlighting the model's strong potential for lightweight deployment.

Key Words: Cross-day neural decoding, Continuous-time modeling, Energy-efficient decoding

1 Introduction

Brain–computer interfaces (BCIs) aim to decode neural activity to control external devices and have demonstrated substantial potential in motor rehabilitation[1], assistive control, and neuroscience research. A central challenge, however, is the pronounced degradation of cross-day decoding performance: models that perform well on the training day often exhibit a rapid decline in accuracy on subsequent recording days. This degradation typically arises from the high variability inherent in neural recordings, causing the statistical distribution of neural activity to drift continuously over multiple days. Developing stable and efficient cross-day decoders without relying on extensive labeled data is therefore a critical prerequisite for practical BCI systems.

Conventional cross-day decoding approaches predominantly rely on deep artificial neural networks (ANNs), such as convolutional networks or [2]. While these methods achieve satisfactory short-term performance, they are often unstable when faced with long-term distributional shifts. Moreover, their dense computation paradigm not only limits adaptation to the inherently sparse nature of neural signals, but also imposes significant energy costs, which hinders their application in real-time BCI systems. Consequently, improving cross-day stability while reducing computational overhead has become a key focus in BCI decoding research.

Spiking neural networks (SNNs) offer a modeling paradigm that is naturally suited to dynamic neural signals. Compared with conventional ANNs[3], SNNs exploit event-driven spiking representations to capture the sparse temporal structure of neural activity, thereby reducing redundant computation and enhancing temporal sensitivity. Nevertheless, existing SNNs exhibit limited capacity for modeling high-dimensional neural signals, particularly in capturing long-range temporal dependencies, which restricts their direct application to cross-day decoding tasks.

In this work, we propose a novel framework for cross-day BCI decoding. Neural activity is directly encoded into spikes, forming sparse, event-driven temporal sequences that enable the model to focus on transient neural events while mitigating distributional shifts caused by inter-day firing rate variations. A Spiking Transformer is employed to capture cross-channel and cross-time dependencies, while an ODE-LSTM module models the continuous evolution of hidden states to enhance adaptability to cross-day neural dynamics. Experimental results demonstrate that the proposed framework achieves superior cross-day decoding stability and energy efficiency compared with conventional deep learning models.

2 Related Works

2.1 Lightweight Spiking Neural Networks (SNNs) and Sparse Representations

Spiking neural networks (SNNs)[4], inspired by the event-driven computation of biological neurons, employ binary spike firing to realize sparse and energy-efficient processing. Compared with traditional artificial neural networks (ANNs), SNNs possess inherent advantages in hardware realization, temporal information processing, and power consumption, making them highly attractive for neural decoding and modeling of temporal neural signals[5]. To further enhance their deployability in practical scenarios—particularly in online BCIs—recent studies have increasingly focused on lightweight design and sparse representation strategies.

This work was supported by projects 2024YFF1400600 and 2024YFF1400604. Corresponding author: Tielin Zhang.

One line of work improves the trainability of SNNs by leveraging surrogate gradients together with event-driven sparsification mechanisms, enabling the network to maintain high representational capacity while substantially reducing inference energy. Another line of research introduces structural pruning, channel sparsification, weight sharing, and quantization techniques, which compress model size from an architectural perspective. For example, SNNs designed with multi-threshold membrane potential accumulation and time-step sparsification greatly reduce neuronal firing rates, thereby lowering computational complexity.

Recently, with the success of the Transformer framework in sequence modeling, researchers have begun exploring the integration of SNNs with Transformer architectures, giving rise to Spike Transformers[6]. These models typically employ event-driven sparse self-attention, where attention operations are triggered only upon the occurrence of spike events, further reducing computational [7]. Spike Transformers thus retain the powerful sequence modeling capabilities of Transformers while preserving the biologically inspired sparse computation of SNNs, providing a promising path toward achieving both high expressiveness and lightweight computation for cross-day neural decoding.

2.2 Continuous-time modeling based on ODE

Due to the high variability inherent in neural recordings, deep neural networks that rely on discrete-time modeling struggle to accommodate statistical shifts occurring across multiple days. Continuous-time neural networks based on ordinary differential equations (ODEs) provide a new perspective for modeling such dynamic systems[8]. Neural ODEs interpret the evolution of a network's hidden states as the numerical solution of an ODE, thereby enabling temporally continuous and numerically stable state updates. Building on this foundation, models such as ODE-LSTM and CT-LSTM integrate ODE formulations with recurrent memory units, allowing temporal dependencies to be preserved within a continuous dynamical framework[9]. These architectures demonstrate enhanced robustness under irregular sampling conditions and long-term drift scenarios.

3 Proposed Method

This section provides an overview of the proposed unified framework for spatiotemporal spike-based sequence modeling, which is composed of three tightly integrated components: Direct Spiking Encoding, the Spiking Transformer for Spatiotemporal Modeling, and the ODE-LSTM for Continuous-time Decoding. These modules operate synergistically, enabling the system to transform raw continuous inputs into spike-based representations and subsequently reconstruct continuous-time dynamics in an end-to-end manner. The overall architecture is illustrated in Fig.1.

3.1 Direct Spiking Encoding

Traditional SNNs typically rely on an additional encoding stage—such as Poisson encoding, temporal encoding, or amplitude encoding—to convert continuous-valued pixels into spike trains. This procedure not only introduces extra computational overhead but also results in information loss and latency. To circumvent these limitations, we feed the feature maps generated by the first convolutional layer (Conv1) directly into the spiking neuron layer, thereby enabling a seamless transition from continuous signals to spike-based processing. Specifically, the input image is first processed by a standard convolutional layer to extract local spatial features, which are subsequently passed into multi-step LIF/PLIF neurons. Under this design, the spiking neurons no longer depend on a separate encoder to produce spike sequences; instead, they autonomously discretize the continuous activations through membrane potential dynamics, achieving an inherent, end-to-end conversion from the pixel domain to the spiking domain.

3.2 Spiking Transformer for Spatiotemporal Modeling

To efficiently model spatiotemporal dependencies in time-series data, we develop a Spiking Transformer (ST) aligner that integrates the discrete event-driven dynamics of spiking neurons with the sequence modeling capability of Transformers. This design enables the model to inherit the energy-efficient binary computation of SNNs while retaining the capacity to capture long-range temporal dependencies. Given a multi-channel time-series input $X \in \mathbb{R}^{B\times C\times L}$, the model first applies a multi-scale convolutional Patch Embedding module to partition the sequence into local temporal blocks and project them into a unified feature space, as follows:

$$Z_0 = \mathrm{PatchEmbed}(X) = \phi(W * X + b), \qquad (1)$$

where ϕ denotes the binary spike sequences generated by the first-layer multi-step LIF neurons.

To ensure numerical stability throughout the network depth, we introduce a SpikeRescale(SR) module after every LIF layer. This module dynamically adjusts the firing rate and membrane potential magnitude so that activations remain within an appropriate range during multi-layer propagation. Give the LIF output spike sequence $s_t \in \{0, 1\}$, SR modulates the membrane potential using learnable scaling and shifting parameters γ and β,as follows:

$$\tilde{s}_t = \gamma(s_t - p) + \beta, \qquad (2)$$

where γ and β are learnable affine parameters, and p denotes the estimated mean firing probability serving as a reference for normalization.

This operation is analogous to normalization but preserves the discreteness of spikes and therefore does not compromise the event-driven nature of the network; it can be regarded as a normalization strategy defined in the spike domain. Its role is crucial in deep propagation, as the membrane-potential accumulation mechanism of LIF neurons can cause the firing rate to drift across layers. SpikeRescale continuously compensates for this drift, ensuring that the attention and FFN modules in deeper layers operate within a numerically stable range.

The model then enters the backbone stage, which is composed of multiple stacked Transformer blocks built upon spiking attention. Unlike conventional Transformers, the $Q/K/V$ projections are each passed through LIF neurons to produce binary spike representations, while SpikeRescale compensates for channel-wise variations in firing intensity after projection, ensuring that the attention scores are not in-

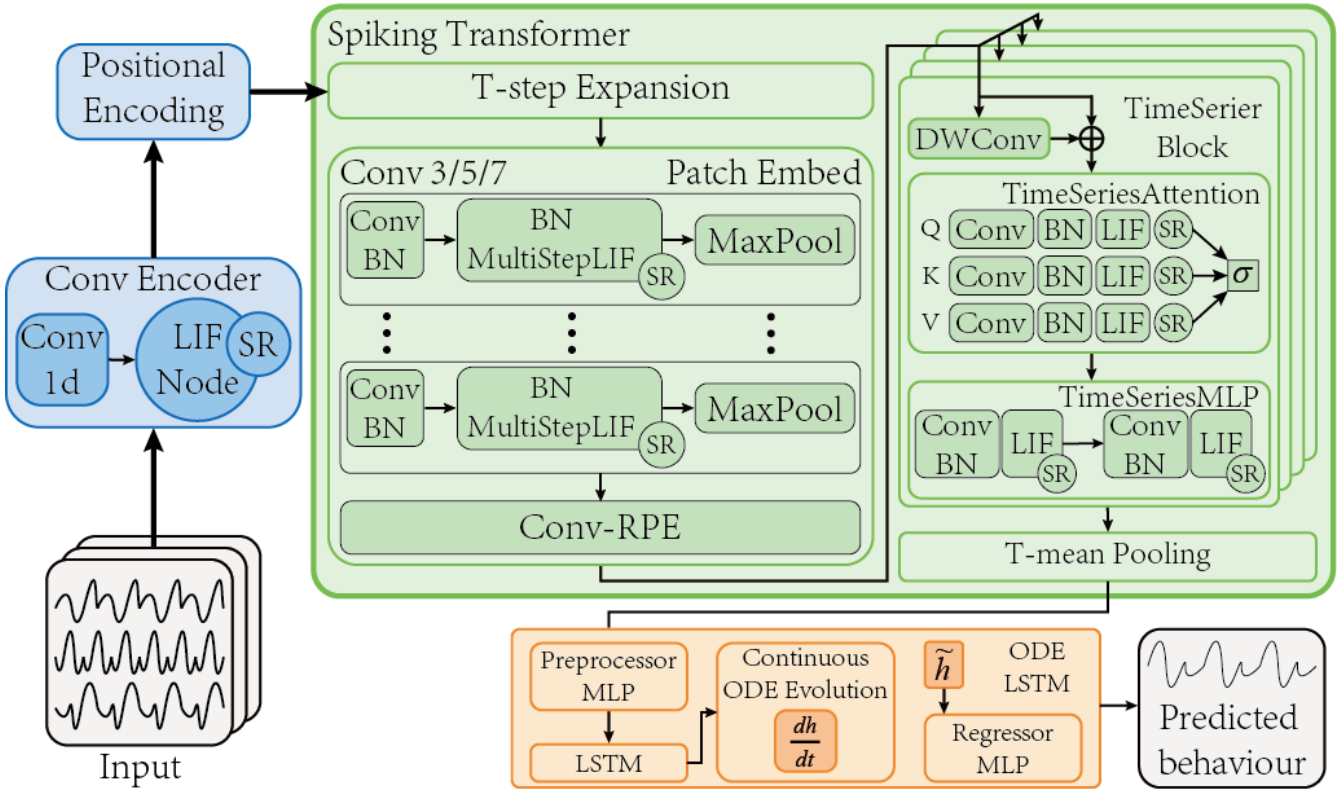

Fig. 1: The proposed unified framework for spatiotemporal spiking sequence modeling comprises three primary components: a Direct Spiking Encoding module, a Spiking Transformer for Spatiotemporal Modeling, and a Continuous-time Dynamic Decoder based on ODE-LSTM. Together, these modules establish an integrated architecture that effectively transforms raw continuous inputs into spike-based representations, models spatiotemporal dependencies, and enables continuous-time prediction.

fluenced by abnormally amplified activations,as follows:

$$\begin{aligned} Q &= \mathrm{SR}(S_q(W_q Z)), \\ K &= \mathrm{SR}(S_k(W_k Z)), \\ V &= \mathrm{SR}(S_v(W_v Z)). \end{aligned} \tag{3}$$

where W_q, W_k, W_v are learnable projection matrices, S_q, S_k, S_v denote the LIF neurons applied to each projection branch,

The output after computing the attention map is again passed through a LIF neuron followed by SR module, ensuring that the entire attention pathway maintains dynamically stable activation scales.

To enhance the modeling of short-term temporal dependencies, we introduce a TokenMix operation before the attention module, enabling local temporal mixing of the sequence prior to self-attention. The output of this stage is also processed by LIF and SR modules to maintain a consistent activation scale before entering the attention block. Subsequently, the SFFN component is implemented as a spiking MLP, as follows:

$$\mathrm{SFFN}(Z) = \mathrm{SR}\left(S_2\left(W_2\,\mathrm{SR}(S_1(W_1 Z))\right)\right) \tag{4}$$

The entire Transformer block integrates temporal information through the combination of spiking attention and a spiking FFN(SFFN), and is trained stably through residual connections:

$$Z_{l+1} = Z_l + \mathrm{SFFN}(\mathrm{SpikeAttn}(Z_l)) \tag{5}$$

where $SpikeAttn(\cdot)$ denotes the spiking self-attention operator combining binary $Q/K/V$ projections with SR-stabilized attention maps.

Finally, at the output stage of the aligner, the network performs statistical pooling over the spike features across all time steps to integrate the temporal information, as follows:

$$\bar{Z} = \frac{1}{T}\sum_{t=1}^{T} s_t \tag{6}$$

where s_t denotes the spike vector at time step t, and $\bar{Z}$ is the temporally aggregated representation used for downstream decoding.

3.3 ODE-LSTM for Continuous-time Decoding

To achieve robust recovery of neural sequence dynamics in the continuous-time domain, we introduce a hybrid temporal unit that integrates ordinary differential equation (ODE) modeling with a recurrent memory architecture, termed ODE-LSTM. The central idea of this module is to augment the conventional gating-based memory mechanism of an LSTM with an ODE-governed evolution of the hidden states, enabling inference of system dynamics at arbitrary continuous time points. This relaxes the constraint of fixed sampling intervals and facilitates continuous-time modeling and high–temporal-resolution decoding of sequential signals. Let the input neural sequence be denoted as $x(t)$, and the hidden and cell states of the LSTM be represented by $h(t)$ and $c(t)$, respectively. In discrete-time networks, these

Algorithm 1 Aligner SpikeTransformer

Require: sequence $X \in \mathbb{R}^{L\times B\times C_{in}}$
Ensure: prediction $Y \in \mathbb{R}^{L\times B\times C_{out}}$
function INITIALIZE($input\_dim$)
  $PosEnc \leftarrow PositionalEncoding(d\_model)$
  $ConvEnc \leftarrow Conv1d(input\_dim, 128, k{=}3, pad{=}1) \rightarrow LIF(\tau{=}2, v\_th{=}1)$
  $SNN\_Tr \leftarrow TimeSeriesSpikeTransformer(T{=}4)$
  $Dec \leftarrow Linear(d\_model, ntoken)$
  return $(PosEnc, ConvEnc, SNN\_Tr, Dec)$
end function
function FORWARD(X)
  $X \leftarrow permute(X, [B, C_{in}, L])$
  $X \leftarrow ConvEnc(X)$ ▷ spike encoding
  $X \leftarrow permute(X, [L, B, C_{mid}])$
  $X \leftarrow PosEnc(X)$
  $X \leftarrow permute(X, [B, C_{mid}, L])$
  $Z \leftarrow SNN\_Tr(X)$
  $Y \leftarrow Dec(Z)$
  reset_spike_states()
  return Y
end function
procedure TRAIN($data\_loader$)
  for (X, T_gt) **in** $data\_loader$ **do**
    $Y \leftarrow Forward(X)$
    $loss \leftarrow Loss(Y, T\_gt)$
    optimizer.zero_grad()
    loss.backward()
    optimizer.step()
  end for
end procedure

states can only be updated at predefined sampling steps Δt; in contrast, ODE-LSTM treats them as continuous-time variables whose trajectories evolve according to a neural ODE of the form:

$$\frac{dh(t)}{dt} = f_0(h(t), t) \tag{7}$$

where $h(t)$ is the continuous hidden state, and $f_0(\cdot)$ is a differentiable dynamic function implemented by a multilayer perceptron (MLP), which characterizes the continuous evolution of the underlying neural dynamics.

To obtain hidden-state predictions within an interval $[t_k, t_{k+1}]$, we integrate the above differential equation using a numerical ODE solver—such as Euler, Runge–Kutta (RK4), or an adaptive step-size integrator—yielding the continuous-time trajectory of the latent state, as follows:

$$\tilde{h}_{t_{k+1}} = h_{t_k} + \int_{t_k}^{t_{k+1}} f_0(h(t), t)\, dt \tag{8}$$

where $\tilde{h}_{t_{k+1}}$ denotes the ODE-propagated hidden state at time t_{k+1}, h_{t_k} is the hidden state at the previous observation time t_k. This integration implies that, before the LSTM receives the next input, its hidden state has already evolved according to the underlying dynamics, thereby enabling a form of continuous-time prediction.

After the ODE-based propagation, the network performs a standard LSTM gating update to incorporate the incoming input and correct the prediction error, as follows:

$$\begin{aligned} f_t &= \sigma(W_f x_t + U_f \tilde{h}_t + b_f), \\ i_t &= \sigma(W_i x_t + U_i \tilde{h}_t + b_i), \\ o_t &= \sigma(W_o x_t + U_o \tilde{h}_t + b_o), \end{aligned} \tag{9}$$

where f_t, i_t, o_t are the forget, input, and output gates at time t, $\sigma(\cdot)$ is the sigmoid nonlinearity; and $\tilde{h}_t$ is the ODE-evolved hidden state used as the recurrent input to the gates.

The candidate cell state is formulated as follows:

$$\tilde{c}_t = \tanh(W_c x_t + U_c \tilde{h}_t + b_c) \tag{10}$$

Accordingly, the cell state and hidden state are updated as follows:

$$\begin{aligned} c_t &= f_t \odot c_{t^-} + i_t \odot \tilde{c}_t, \\ h_t &= o_t \odot \tanh(c_t). \end{aligned} \tag{11}$$

where c_{t^-} denotes the internal state after ODE-based evolution but prior to incorporating the current input, c_t is the updated cell state, h_t is the updated hidden state, and $\odot$ denotes element-wise multiplication.

Algorithm 2 ODE-LSTM Decoder (Compact)

Require: $X \in \mathbb{R}^{B\times T\times C_{in}}$, timespans $\in \mathbb{R}^{B\times T}$, optional mask
Ensure: $Y \in \mathbb{R}^{B\times T\times C_{out}}$ or $\mathbb{R}^{B\times C_{out}}$
function INITIALIZE($input\_dim, hidden\_size, num\_layers$)
  Initialize $h_\ell, c_\ell \leftarrow 0$ for $\ell = 1 \dots L$
  Linear output layer: $Linear(hidden\_size, out\_feature)$
  return $(h, c, Linear)$
end function
function FORWARD($X, timespans, mask$)
  outputs $\leftarrow \emptyset$, last_output $\leftarrow 0$
  for $t = 1 \dots T$ **do**
    $x_t \leftarrow X[:, t]$, $dt_t \leftarrow$ timespans[:,t]
    for $\ell = 1 \dots L$ **do**
      $x_{\text{in}} \leftarrow x_t$ if $\ell = 1$ else $h_{\ell-1}$
      $(h_\ell, c_\ell) \leftarrow$ LSTMCell$(x_{\text{in}}, (h_\ell, c_\ell))$
      $h_\ell \leftarrow$ FixedStepSolve(h_ℓ, dt_t)
      if dropout and $\ell < L$ **then**
        $h_\ell \leftarrow$ Dropout(h_ℓ)
      end if
    end for
    $y_t \leftarrow Linear(h_L)$, append outputs
    last_output $\leftarrow$ mask[:,t]*y_t + (1-mask[:,t])*last_output if mask else y_t
  end for
  $Y \leftarrow$ stack(outputs) if return_sequences else last_output
  return Y
end function
function FIXEDSTEPSOLVE(h, dt)
  $dt_s \leftarrow dt/3$
  for $i = 1 \dots 3$ **do**
    $h \leftarrow$ ODE_Step(h, dt_s)
  end for
  return h
end function
function ODE_STEP(y, dt)
  return $y + dt * f(y)$ ▷ Euler / Heun / RK4 unified representation for simplicity
end function

Through this dual-stage update mechanism—ODE-driven evolution followed by LSTM-based correction—the model

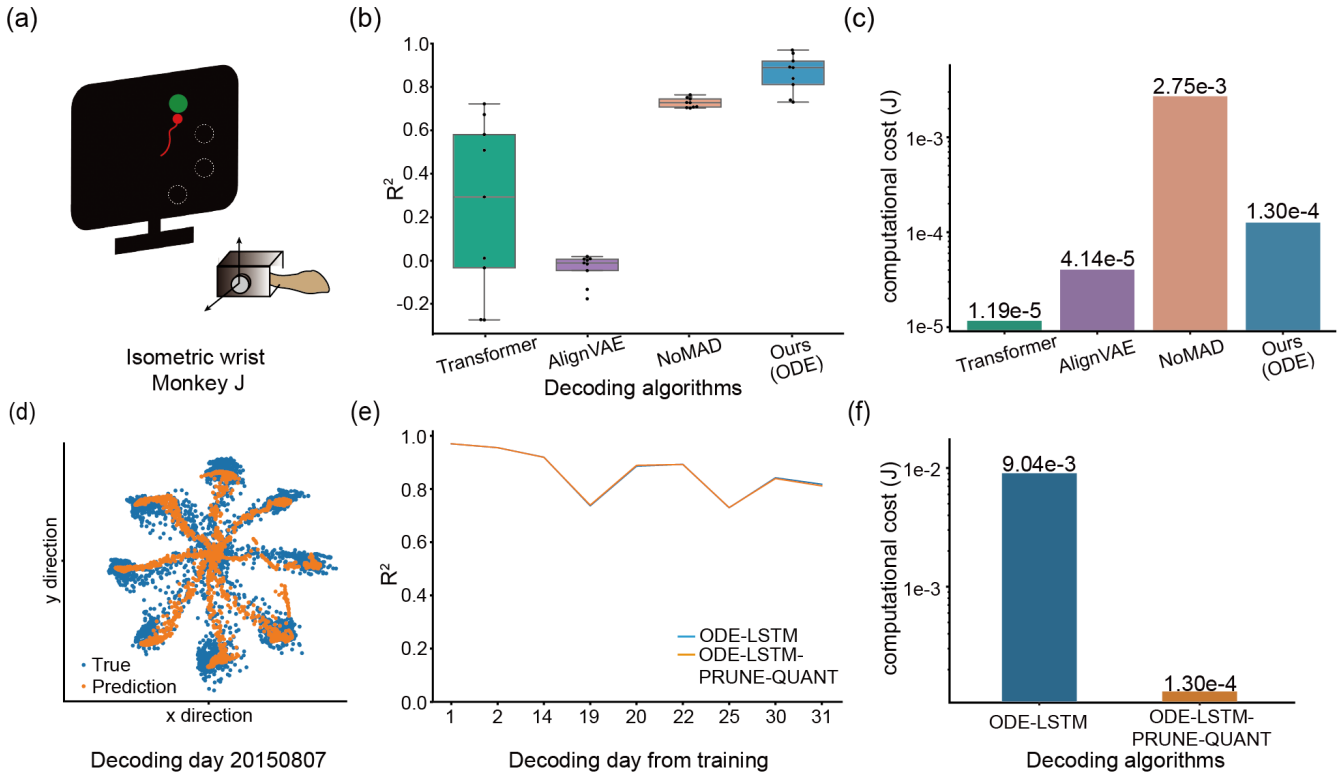

Fig. 2: Results on the isometric wrist task for Monkey J. (a) Experimental paradigm of the isometric wrist force control task. (b)-(c) Cross-day decoding performance of different methods, including Transformer[10], AlignVAE[11] and NoMAD[12], along with their corresponding theoretical energy consumption. (d) Eight-direction sequence prediction on a single day. (e)–(f) Cross-day decoding performance and theoretical energy consumption comparison between the original model and its lightweight variant after pruning and quantization.

preserves the strong representational capacity of the LSTM while naturally adapting to irregular sampling patterns, cross–time-scale variability, and the continuous dynamical characteristics of neural encoding. Importantly, ODE-LSTM is not merely an RNN augmented with an external ODE solver; rather, it interprets the internal LSTM states as discrete observations of an underlying continuous-time dynamical system, thereby enabling the model to track the neural activity manifold in continuous time. Because the hidden state $h(t)$ is governed by the aforementioned differential equation, its value at any intermediate time point $t \in (t_k, t_{k+1})$ can be obtained directly via the solver. This property is particularly advantageous in long-term neural decoding, modeling of non-uniformly sampled neural data, and multimodal temporal alignment. Moreover, by parameterizing the dynamical function f_θ, the model can automatically learn the intrinsic evolution rules of neural activity—such as slow drifts, state-dependent transitions, or latent oscillatory structures—which conventional discrete-time LSTM architectures often struggle to capture, ultimately leading to improved predictive performance.

4 Experimental Data and Analysis

We evaluate the proposed framework in terms of its cross-day generalization capability, as well as its theoretical energy consumption on two invasive brain–computer interface (BCI) datasets. The results confirm both the effectiveness of the proposed method and its potential practical applicability.

4.1 Datasets

We evaluate the proposed method on two invasive brain–computer interface (BCI) datasets corresponding to different motor task paradigms and neural recording configurations.

Isometric Wrist Task: As shown in Fig.2(a), in the isometric wrist task the hand of Monkey J was fixed in a stationary position, and cursor control was achieved through forces generated at the wrist. The neural recordings span from July 30, 2015 to November 2, 2015. Due to various factors affecting data quality during long-term recording, only data collected between July 30, 2015 and September 6, 2015 were used in the final experiments.

Center-Out Reaching Task: As illustrated in Fig.3(a), in the center-out reaching task neural signals were recorded from the contralateral primary motor cortex (M1) while monkey D controlled a joystick to move a cursor toward one of eight peripheral target locations. The recordings for monkey D were continuously collected from April 1, 2025 to April 29, 2025.

4.2 Cross-day Decoding Performance and Energy Efficiency

4.2.1 Test on Monkey J

As shown in Fig.2(b), we compare the decoding performance of several representative algorithms on the isometric wrist task using the coefficient of determination (R^2) as the evaluation metric. The results demonstrate that the

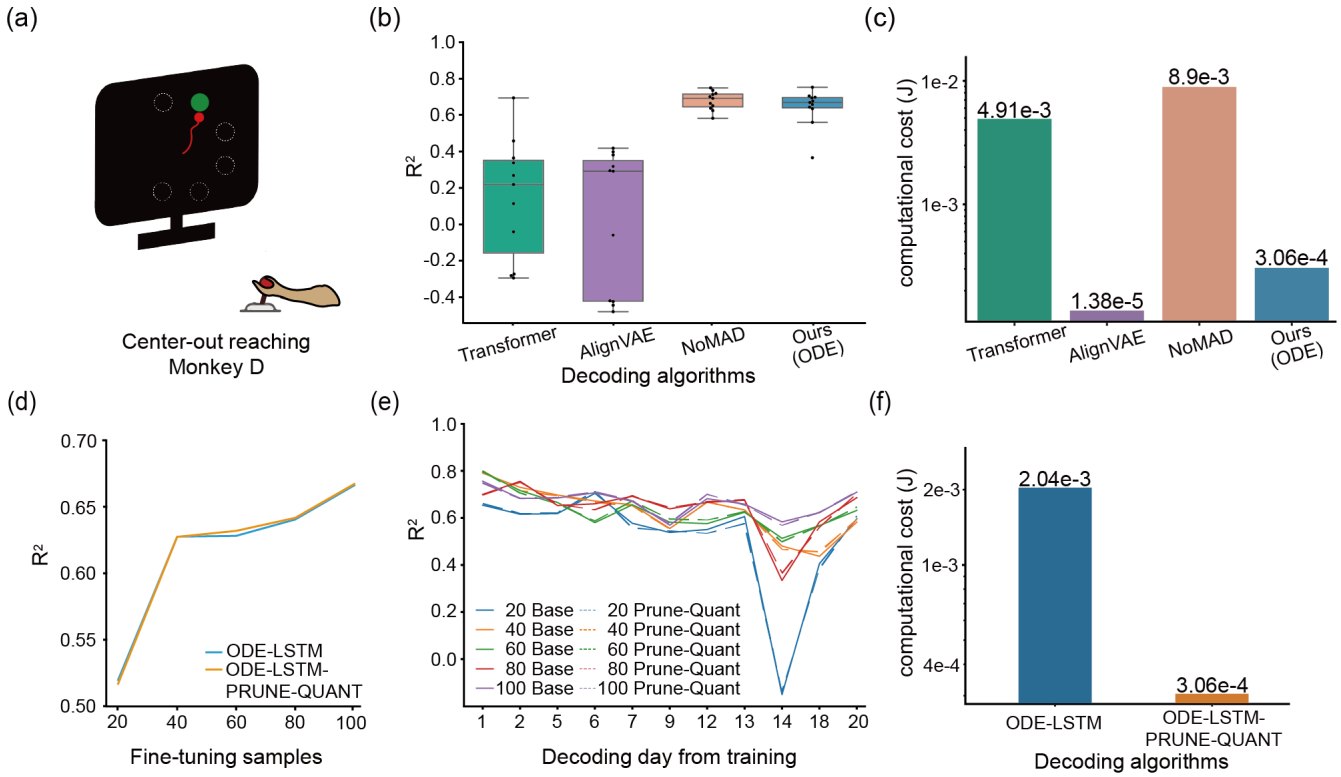

Fig. 3: (a) Experimental paradigm of the isometric wrist force control task. (b)–(c) Cross-day decoding performance of different methods and their corresponding theoretical energy consumption. (d) Model performance with varying amounts of supervised fine-tuning data. (e)–(f) Cross-day decoding performance and theoretical energy consumption of the original model and its lightweight version after pruning and quantization.

proposed continuous-time modeling framework consistently achieves the best overall performance and stability: both the median and upper quartile of the R^2 distribution are markedly higher than those of the baseline methods, with a more compact spread. These findings substantiate that introducing continuous-time dynamical constraints via an ODE solver effectively mitigates performance degradation caused by cross-day neural signal drift, thereby enabling more robust neural decoding.

As shown in Fig.2(c), we further compare the energy consumption of different decoding algorithms during training and evaluation. The proposed model not only achieves the highest decoding accuracy on this dataset but also exhibits the lowest energy cost, highlighting an advantageous balance between performance and efficiency.

Moreover, as illustrated in Fig.2(e), ODE-LSTM maintains high decoding accuracy over an extended temporal span, with only minor performance drops at a few time points and an overall stable trend. This behavior indicates that the model successfully captures the continuous-time evolution of neural activity and demonstrates strong adaptability to cross-day distributional shifts. In addition, the compressed ODE-LSTM-PRUNE-QUANT model exhibits performance curves that closely match those of the original model, with the R^2 values nearly overlapping on most testing days.

The energy evaluation presented in Fig.2(f) shows that the original ODE-LSTM incurs substantially higher energy consumption compared with its lightweight counterpart, whereas ODE-LSTM-PRUNE-QUANT achieves orders-of-magnitude reductions in energy cost while preserving decoding performance. Collectively, these results indicate that the learned continuous dynamical representations possess considerable redundancy and stability, thereby supporting substantial model simplification without compromising cross-day generalization capability.

4.2.2 Test on Monkey D

Given the higher behavioral variability and neural signal complexity of the Monkey-D dataset, we introduce a small amount of supervised fine-tuning in the cross-day evaluation to compensate for distribution shifts across recording days. Fig.3(b) compares the cross-day decoding performance of several baseline methods and our model when only 3 minutes of data are used for fine-tuning. Consistent with the results on the wrist-force task, the discrete-time Transformer exhibits substantial performance fluctuations under cross-day evaluation, while the decoding performance of AlignVAE remains limited. NoMAD achieves relatively stable performance by modeling latent neural dynamics; however, our continuous-time ODE model attains comparable overall accuracy and stability, with a more compact and consistent performance distribution. Furthermore, the energy consumption comparison in Fig.3(c) shows that the proposed model achieves significantly better energy efficiency than the competing approaches.

As shown in Fig.3(d), increasing the amount of supervised fine-tuning data leads to diminishing performance improvements for both the original ODE-LSTM and its lightweight

Table 1: Ablation study on different aligner and decoder designs under 9-day cross-day testing. Performance is reported using the coefficient of determination (R^2).

Aligner	**Decoder**	Day_1	Day_2	Day_{14}	Day_{19}	Day_{20}	Day_{22}	Day_{25}	Day_{30}	Day_{31}	**Avg**
Spike-Transformer	LSTM	0.9355	0.9074	0.8706	0.7425	0.8489	0.8055	0.7235	0.5948	0.6792	0.7898
Spike-Transformer	ODE-LSTM	0.9513	0.9234	0.9094	0.6324	0.8428	0.8091	0.7182	0.6014	0.7457	0.7926
Spike-Transformer (MR)	ODE-LSTM	0.9699	0.9554	0.9197	0.7367	0.8858	0.8928	0.7296	0.8423	0.8179	0.8611
Spike-Transformer (MR)	ODE-LSTM-PQ	0.9701	0.9554	0.9191	0.7395	0.8891	0.8923	0.7302	0.8392	0.8117	**0.8607**

version, ODE-LSTM-PRUNE-QUANT, once the sample size reaches approximately 60–80 trials. Therefore, we adopt 80 trials as the standard fine-tuning setting. Notably, the lightweight model achieves almost identical performance to the original model across all data scales. Fig.3(e) presents the cross-day decoding results under different fine-tuning sample sizes, demonstrating that both ODE-based models maintain high and stable decoding accuracy over extended temporal spans. Combined with Fig.3(f), these results further validate the effectiveness of our model's lightweight design.

Taken together, these findings confirm that continuous-time neural dynamics modeling provides an efficient, stable, and reliable decoding framework for real-world BCI applications.

4.3 Ablation experiment

To systematically evaluate the impact of different module designs on cross-day neural decoding, we conduct a series of ablation studies on the isometric wrist force control task, with the results reported in Table 1. Specifically, we examine the contributions of the aligner architecture, decoder formulation, and model compression strategy. Under the same Spike-Transformer aligner, replacing the conventional LSTM with ODE-LSTM improves the mean R^2 from 0.7898 to 0.7926. Although the overall gain is modest, ODE-LSTM yields more stable performance on later test days, indicating that continuous-time dynamical modeling helps alleviate performance degradation over long cross-day intervals. Furthermore, equipping the ODE-LSTM decoder with a Multi-scale Convolution (MR)-enhanced Spike-Transformer aligner substantially improves cross-day generalization, boosting the average R^2 to 0.8611. This result suggests that multi-scale representation learning enables the aligner to capture neural dynamics at different temporal scales and thus achieve more robust cross-day feature alignment. Building on this stronger architecture, we further develop a lightweight ODE-LSTM through structured pruning and post-training quantization (PQ). Despite a significant reduction in model complexity, the compressed model still achieves a competitive average R^2 of 0.8607, indicating that the learned continuous-time neural dynamical representations contain considerable redundancy and can preserve cross-day decoding performance under substantial parameter and computational compression.

5 Conclusion

This work proposes a unified framework for cross-day BCI decoding by integrating direct spike-based encoding, Spiking Transformer spatiotemporal modeling, and ODE-LSTM continuous-time decoding. By jointly modeling neural signal representation, temporal dependencies, and latent-state dynamics, the proposed method improves decoding stability, robustness, and energy efficiency across multiple recording days. Experimental results show that the framework outperforms existing deep learning methods in cross-day performance, computational cost, and decoding accuracy. This study provides a promising direction for developing long-term deployable and low-power BCI decoding systems, and it also has the potential to be extended to broader neural engineering applications, such as closed-loop control and cross-modal neural information integration.

References

[1] X. Tang, H. Shen, S. Zhao, N. Li, and J. Liu, "Flexible brain–computer interfaces," *Nature Electronics*, vol. 6, no. 2, pp. 109–118, 2023.

[2] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI open*, vol. 3, pp. 111–132, 2022.

[3] K. Yamazaki, V.-K. Vo-Ho, D. Bulsara, and N. Le, "Spiking neural networks and their applications: A review," *Brain sciences*, vol. 12, no. 7, p. 863, 2022.

[4] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, "Deep learning in spiking neural networks," *Neural networks*, vol. 111, pp. 47–63, 2019.

[5] H.-Y. Lu, E. S. Lorenc, H. Zhu, J. Kilmarx, J. Sulzer, C. Xie, P. N. Tobler, A. J. Watrous, A. L. Orsborn, J. Lewis-Peacock *et al.*, "Multi-scale neural decoding and analysis," *Journal of neural engineering*, vol. 18, no. 4, p. 045013, 2021.

[6] Y. Li, Y. Lei, and X. Yang, "Spikeformer: Training high-performance spiking neural network with transformer," *Neurocomputing*, vol. 574, p. 127279, 2024.

[7] N. Rathi and K. Roy, "Diet-snn: A low-latency spiking neural network with direct input encoding and leakage and threshold optimization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 6, pp. 3174–3182, 2021.

[8] A. Fedorov, A. Perechodjuk, and D. Linke, "Kinetics-constrained neural ordinary differential equations: Artificial neural network models tailored for small data to boost kinetic model development," *Chemical Engineering Journal*, vol. 477, p. 146869, 2023.

[9] J. Wang, J. Li, X. Wang, J. Wang, and M. Huang, "Air quality prediction using ct-lstm," *Neural Computing and Applications*, vol. 33, no. 10, pp. 4779–4792, 2021.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[11] A. Vermani, I. M. Park, and J. Nassar, "Leveraging generative models for unsupervised alignment of neural time series data," in *The Twelfth International Conference on Learning Representations*, 2023.

[12] B. M. Karpowicz, Y. H. Ali, L. N. Wimalasena, A. R. Sedler, M. R. Keshtkaran, K. Bodkin, X. Ma, D. B. Rubin, Z. M. Williams, S. S. Cash *et al.*, "Stabilizing brain-computer interfaces through alignment of latent dynamics," *Nature Communications*, vol. 16, no. 1, p. 4662, 2025.