Research Proposal: KAN-MAMOTE for Adaptive Spatio-Temporal Representation Learning in Continuous-Time Dynamic Systems

[Your Name]
[Your Affiliation/Department]

Submitted: [Current Date]

Abstract

Modeling continuous-time dynamic systems, where interactions and states evolve asynchronously at arbitrary timestamps, presents profound challenges for machine learning. Existing time encoding methods, including recent advancements like Learnable Transformation-based Generalized Time Encoding (LeTE), often rely on pre-defined families of basis functions (e.g., Fourier series, splines). While effective, this inherent inductive bias limits their capacity to precisely capture the full spectrum of complex, irregular, non-stationary, or uncharacterized temporal dynamics. Furthermore, integrating these static encodings with sequential memory models for continuous time remains a non-trivial task.

We propose **KAN-MAMOTE (Kernel-Adaptive-Neural-Mamba-Mixture-of-Time-Experts)**, a novel, comprehensive, and theoretically grounded framework designed to overcome these limitations. KAN-MAMOTE comprises two synergistic modules:

(i) **K-MOTE (Kernel-Mixture-of-Time-Experts):** An adaptive time encoding module that leverages a **Mixture-of-Experts (MoE)** architecture. A dynamic router intelligently selects and combines specialized learnable basis function experts: an **Advanced Fourier-KAN Expert** (with trainable, non-integer frequencies), a **Spline-KAN Expert** (leveraging FasterKAN's efficiency and MatrixKAN-like parallelization), a **Parameterized RKHS/GaussianKernel KAN Expert** (for data-driven discovery of optimal, flexible bases), and a **Wavelet-KAN Expert** (for localized time-frequency representation).

(ii) **Continuous-Mamba Integration:** A novel mechanism that seamlessly processes sequences of K-MOTE-encoded absolute and relative time embeddings using a **Continuous-Time Mamba** block, dynamically adapting its state transitions to account for irregular time differences between events.

KAN-MAMOTE operates in a plug-and-play manner, automatically learning the most suitable type of non-linear transformation for any given timestamp and its sequential context. This approach promises unprecedented adaptability, superior performance on complex irregular data, enhanced interpretability, and reduced reliance on manual hyperparameter tuning, setting a new standard for spatio-temporal representation learning in continuous-time dynamic systems.

1 Introduction

The analysis of continuous-time dynamic systems, ranging from financial markets and social networks to biological processes and climate modeling, is fundamental across scientific and engineering disciplines. In these systems, events and states evolve asynchronously at arbitrary, continuous timestamps, presenting a formidable challenge for traditional machine learning models. Effectively learning robust and expressive representations of temporal information is paramount for critical tasks such as event forecasting, anomaly detection, and dynamic link prediction.

Existing time encoding methodologies, while advancing in sophistication, grapple with inherent limitations:

(1) **Fixed Inductive Biases:** Many methods, from Hand-Crafted Time Encodings (HCTE) to Functional Time Encodings (FTE), impose rigid, predefined functional forms (e.g., fixed trigonometric functions). Even advanced approaches like LeTE [?], which make transformation functions learnable, still constrain these functions to preselected families of basis functions (e.g., Fourier series or B-splines). This inherent bias limits their capacity to precisely capture the full spectrum of complex, irregular, non-stationary, or uncharacterized temporal dynamics prevalent in real-world data.

(2) **Suboptimal for Diverse Temporal Patterns:** Real-world temporal data exhibits a rich tapestry of patterns: global periodicities (e.g., daily cycles), local non-periodic trends (e.g., a sudden increase in activity), transient events (e.g., a brief spike), and long-term memory effects. No single fixed basis function type is optimally efficient or expressive for all these phenomena.

(3) **Manual Hyperparameter Dependence:** Approaches combining different function types often require manual tuning of mixing hyperparameters (e.g., p in LeTE's Combined LeTE), which are dataset- and task-dependent, increasing development overhead.

(4) **Discretization Gap in Sequential Models:** While sequential models like Recurrent Neural Networks (RNNs) and State-Space Models (SSMs) excel at processing sequences, their fundamental discrete-time nature struggles with irregularly sampled continuous-time data. Directly feeding static time encodings into these models often fails to account for the actual time elapsed between events, leading to a loss of crucial temporal dynamics.

We propose **KAN-MAMOTE**, a novel, comprehensive, and theoretically grounded framework designed to surmount these limitations. KAN-MAMOTE introduces a paradigm shift by enabling the model to **dynamically select and adapt the *type* of non-linear basis function** most suitable for a given timestamp, and subsequently integrate these rich embeddings into a continuous-time sequential context.

2 Background: Learnable Time Encoding and Function Learning

Functional Time Encoding (FTE): FTEs [?, ?] map scalar time to a high-dimensional embedding using linear transformations followed by fixed non-linearities (e.g., sine). These are widely adopted in dynamic graph representation learning.

Learnable Transformation-based Generalized Time Encoding (LeTE): LeTE [?] generalizes FTE by making the non-linear transformation functions (ϕ_i) learnable. It parameterizes ϕ_i using either Fourier series expansion or B-splines with learnable coefficients. LeTE demonstrates superior empirical performance across diverse domains by capturing periodic, non-periodic, and mixed patterns, proving invariance to time rescaling and offering interpretability.

Kolmogorov-Arnold Networks (KANs) and MLP-KAN: KANs [?] are a new class of neural networks where activations are learnable spline functions on edges, offering superior approximation capabilities for function learning compared to MLPs. MLP-KAN [?] unifies MLPs (for representation learning) and KANs (for function learning) within a **MoE** architecture, using a router to dynamically select experts. This framework is highly relevant as time encoding is fundamentally a function learning task.

Continuous-Time State-Space Models (SSMs) and Mamba: SSMs are powerful models for sequential data. While Mamba [?] is a discrete-time SSM, its underlying principles can be adapted to continuous-time irregular data by dynamically adjusting its discretization parameters based on the actual time differences between events [?, ?]. This allows for modeling continuous temporal evolution and memory.

3 Proposed Method: KAN-MAMOTE Architecture

K-MAMOTE is a holistic framework for learning spatio-temporal representations. It consists of two primary, synergistic modules: K-MOTE for rich, adaptive point-in-time encoding, and a Continuous-Mamba integration for sequential context and memory.

3.1 K-MOTE: Kernel-Mixture-of-Time-Experts

K-MOTE is the core time encoding module. It is designed to be plug-and-play, taking a scalar timestamp (t_k) and producing a D_{time}-dimensional embedding. It achieves its adaptive power through a MoE architecture, where experts specialize in distinct types of learnable function approximations.

3.1.1 MoE Gating Mechanism (Router)

The router is the central intelligence of K-MOTE, dynamically determining the optimal blend of basis function experts for a given input t_k. This directly addresses LeTE's limitation of a fixed mixing hyperparameter (p).

- **Input:** The scalar timestamp t_k. Auxiliary temporal features (e.g., $\Delta t_{u,k}$ (time since last event), or simple statistics of the local time window) can be included to provide contextual information to the router.
- **Method:** A small, efficient Multi-Layer Perceptron (MLP) takes the input and computes logits for each expert: $logits_e = \text{MLP}_{\text{router}}([t_k, \text{auxiliary\_features}])$.

image_27da00.png

Figure 1: Conceptual Architecture of K-MOTE and K-MAMOTE. K-MOTE dynamically selects experts for time encoding. K-MAMOTE integrates K-MOTE outputs into a Continuous-Mamba for sequential context.

- **Dynamic Weighting:** Softmax is applied to these logits to obtain normalized dispatch weights: $\alpha_e = \frac{\exp(logits_e)}{\sum_{e'} \exp(logits_{e'})}$.

- **Efficient Dispatch** $(\mathbf{Top\text{-}K_{top}})$: $InspiredbyMLP - \mathbf{KAN}[?], a$

3.1.2 Specialized Basis Function Experts

Each expert is a learnable non-linear transformation $\phi_e(x)$ that maps a linearly transformed time $x = \omega_d t_k + \varphi_d$ (where ω_d, φ_d are learnable per output dimension d) to a portion of the D_{time}-dimensional output space.

Expert 1: Advanced Fourier-KAN Expert

- **Purpose:** To capture global, multi-scale, and **precisely learned periodic patterns**. It addresses LeTE's limitation of fixed integer harmonic frequencies.
- **Method:** For each output dimension d, the transformation function $\phi_{\text{Fourier},d}(x_d)$ is defined

as:

$$\phi_{\text{Fourier},d}(x_d) = a_{d,0}+\sum_{j=1}^{K'}(a_{d,j}\cos(\lambda_{d,j}x_d)+b_{d,j}\sin(\lambda_{d,j}x_d))$$

Novelty: $\lambda_{d,j}$ are **learnable real-valued frequencies** for each harmonic j and dimension d. Unlike LeTE's Fourier component, which uses fixed integer multiples for k, this allows the model to discover and adapt to non-harmonic or more precise periodicities present in the data.

- **Learnable Parameters:** $a_{d,0}, a_{d,j}, b_{d,j}$ (coefficients), ω_d, φ_d (linear transformation parameters), and $\lambda_{d,j}$ (learnable real frequencies).
- **Benefits:** Offers finer control over periodic components and potentially more parsimonious fits to complex cyclic patterns. Its operations are inherently parallelizable on modern hardware.

Expert 2: Spline-KAN Expert

- **Purpose:** To model local, non-periodic trends, abrupt changes, and irregular temporal "shapes" with high precision and efficiency.
- **Method:** This expert directly integrates the architecture of **FasterKAN** [**?**], known for its strong performance in function approximation. For each output dimension d, the transformation function $\phi_{\text{Spline},d}(x_d)$ is defined by a learnable spline:

$$\phi_{\text{Spline},d}(x_d) = \sum_{j=1}^{M} c_{d,j} B_j(x_d)$$

where $x_d = \omega_d t_k + \varphi_d$, and $B_j(x_d)$ are B-spline basis functions. The implementation utilizes FasterKAN's optimized grid-based and reflectional switch function mechanisms.

- **Learnable Parameters:** $c_{d,j}$ (control points for B-splines), and ω_d, φ_d.
- **Benefits:** Provides robust approximation for continuous functions, effectively capturing complex non-periodic features and localized events. **Computational efficiency is further enhanced by applying MatrixKAN-like parallelization techniques for B-spline calculations** [**?**], converting recursive computations into efficient matrix operations.

Expert 3: Parameterized RKHS/GaussianKernel KAN Expert

- **Purpose:** To discover **highly flexible, data-driven, and potentially non-stationary "basis functions"** for temporal patterns that may not be optimally captured by pre-defined Fourier or Spline forms. This embodies a deeper level of "learning the basis functions themselves."
- **Method (Parameterized Kernel Representation):** For each output dimension d, the transformation function $\phi_{\text{Kernel},d}(t_k)$ is implicitly defined by a *parameterized positive semi-definite kernel* $K_d(t_a, t_b; \theta_d)$.
  - **Learnable Temporal Anchor Points:** This expert maintains a small, fixed set of P **learnable "temporal anchor points"** $\{t_{\text{anchor}_1}, \ldots, t_{\text{anchor}_P}\}$ in the time domain. These act as reference points for kernel evaluations.
  - **Parameterized Kernel Function $K_d(\cdot,\cdot;\theta_d)$:** This is the core. It's a parameterized function whose internal parameters θ_d are learned. A practical choice is a **Gaussian Mixture Kernel**:

$$K_d(t_a, t_b; \theta_d) = \sum_{m=1}^{N_M} w_{d,m} \exp\left(-\frac{(t_a-\mu_{d,m})^2+(t_b-\mu_{d,m})^2}{2\sigma_{d,m}^2}\right.$$

Here, N_M is a fixed number of mixture components (e.g., 5-10). The parameters $\theta_d = \{w_{d,m}, \mu_{d,m}, \sigma_{d,m}\}_{m=1}^{N_M}$ are **learnable**. They can be directly learned or generated by a tiny KAN (hence "GaussianKernel KAN") taking t_k as input, allowing for context-dependent kernel properties.

  - **Ensuring Positive Semi-Definiteness:** By using a sum of inherently positive semi-definite Gaussian kernels with non-negative weights (e.g., using $\exp(\cdot)$ or $\text{softplus}(\cdot)$ for $w_{d,m}$), the resulting mixture kernel is guaranteed to be valid, satisfying Mercer's Theorem [**?**] and Bochner's Theorem [**?**].
  - **Output Projection:** The kernel similarities $K_d(t_k, t_{\text{anchor}_j}; \theta_d)$ form a vector, which is then projected to the output dimension

via a linear layer.

- **Learnable Parameters:** $\{t_{\text{anchor}_j}\}$, and parameters θ_d of the kernel function, and the final linear layer weights.
- **Benefits:** Offers the highest degree of flexibility, allowing the model to adapt to arbitrary, complex, and uncharacterized temporal patterns by learning implicit similarity measures. Its operations are inherently parallelizable on modern hardware.

Expert 4: Wavelet-KAN Expert

- **Purpose:** To capture **transient, non-stationary, and multi-scale localized temporal features** that might be missed or inefficiently represented by global Fourier, piecewise splines, or fixed-width RBFs.
- **Method:** For each output dimension d, the transformation function $\phi_{\text{Wavelet},d}(x_d)$ is defined as a linear combination of learnable wavelet basis functions:

$$\phi_{\text{Wavelet},d}(x_d) = \sum_{j=1}^{N_W} w_{d,j} \Psi_{s_{d,j},\tau_{d,j}}(x_d)$$

  where $x_d = \omega_d t_k + \varphi_d$. $\Psi_{s,\tau}(x) = \frac{1}{\sqrt{s}} \Psi_0 \left(\frac{x-\tau}{s}\right)$ is a chosen mother wavelet (e.g., Daubechies-4 wavelet, preferred for its smoothness and compact support).
- **Novelty:** $s_{d,j}$ (dilation/scale) and $\tau_{d,j}$ (translation/position) for each wavelet j and dimension d are **learnable parameters**, allowing the wavelet basis to adapt its scale and position to the data. $w_{d,j}$ are learnable coefficients.
- **Learnable Parameters:** $w_{d,j}, s_{d,j}, \tau_{d,j}$, and ω_d, φ_d.
- **Benefits:** Provides powerful time-frequency localization, ideal for signals with transient features or non-stationary characteristics. Its operations are inherently parallelizable on modern hardware.

3.2 Continuous-Mamba Integration in K-MAMOTE

K-MAMOTE integrates the rich embeddings from K-MOTE into a sequential context using a Continuous-Mamba block, addressing the challenge of modeling continuous-time dynamics with irregular sampling.

- **Input:** K-MAMOTE takes both the current absolute timestamp t_k and the time difference to the previous event $\Delta t_k = t_k - t_{k-1}$ as inputs.
- **Parallel K-MOTE Embeddings:** Two K-MOTE modules run in parallel:
  (i) $\Phi_{\text{abs}}(t_k)$: K-MOTE encoding of the absolute timestamp.
  (ii) $\Phi_{\text{rel}}(\Delta t_k)$: K-MOTE encoding of the relative time difference.
- **Continuous-Mamba Block:** These two embeddings are concatenated (along with any raw event features) to form a rich input vector u_k for the Mamba block. The Mamba block processes the sequence of these u_k vectors.
  - **Problem Addressed:** Standard Mamba is discrete. Continuous-time events are irregularly sampled.
  - **Solution:** The Mamba block is adapted to account for the continuous time elapsed. Its internal continuous-time SSM parameters (A, B, C, D) are learned. For each step k in the sequence, the discrete update matrices $(\overline{A}_k, \overline{B}_k)$ are dynamically computed using the actual time difference Δt_k. This is typically achieved by making Mamba's internal discretization parameter (Δ) a function of Δt_k (e.g., $\Delta_{\text{Mamba},k} = \text{softplus}(\text{Linear}(\Delta t_k))$).
  - **Update Rule:** The hidden state h_k (representing the sequential memory) is updated as $h_k = \overline{A}_k h_{k-1} + \overline{B}_k u_k$.
- **Output:** The final hidden state h_k from the Continuous-Mamba block, corresponding to t_k, serves as the comprehensive "Absolute-Relative t_k Embedding." This embedding captures both the point-in-time characteristics (from K-MOTE) and its sequential evolution with memory (from Mamba).

3.3 Regularization for Robustness and Interpretability

To ensure the learned functions are smooth, generalize well, and are robust to noise, we integrate principled regularization techniques:

- **Sobolev L_2 Regularization:** Penalizes the L_2 norm of the first and/or second derivatives of each expert's transformation function $\phi_e(x_d)$. This encourages smoothness and prevents overly oscillatory or erratic learned functions, improving generalization.
- **Total Variation L_1 Regularization:** Penalizes the L_1 norm of the function's gradient (or difference between adjacent values in discrete settings). This promotes piecewise constant or piecewise linear solutions, encouraging sparsity in derivatives and allowing for sharp, interpretable transitions while maintaining smoothness in other regions.
- **MoE Specific Load Balancing Loss:** Essential to ensure all experts are utilized and prevent expert "collapse" (where some experts are rarely chosen), promoting a balanced distribution of expertise.

4 Pros and Cons: Comparison to Other Models

4.1 Advantages of KAN-MAMOTE

- **Unprecedented Adaptivity (Beyond LeTE):** Dynamically selects the most appropriate basis function type (Fourier, Spline, Kernel, Wavelet) for each timestamp, overcoming the fixed inductive biases of LeTE and DTKN. This automates the selection of optimal functional forms.
- **Comprehensive Temporal Modeling:** K-MAMOTE handles the full spectrum of temporal patterns: global periodicity (Advanced Fourier), local non-periodic shapes (Spline-KAN), arbitrary data-driven similarities (GaussianKernel KAN), transient events (Wavelet-KAN), and sequential memory with continuous-time evolution (Continuous-Mamba).
- **Reduced Manual Hyperparameter Tuning:** The MoE router automatically learns the optimal blend of experts, significantly reducing the need for manual tuning of mixing proportions (e.g., LeTE's p).
- **Strong Theoretical Foundations:** Built upon rigorous mathematical concepts including RKHS, Bochner's/Mercer's theorems, KANs, MoE, and Continuous-Time SSMs, ensuring principled design.
- **High Potential for Performance & Generalization:** The adaptive nature and specialized experts are expected to yield superior performance on complex, irregular, and non-stationary real-world datasets.
- **Enhanced Interpretability:** The modular design allows for analyzing router decisions (which expert is chosen when) and visualizing the learned functions of individual experts, providing deeper insights into the model's temporal understanding.
- **Plug-and-Play Compatibility:** K-MAMOTE provides a high-dimensional embedding that can seamlessly integrate into existing downstream models (e.g., Transformers, TGNs).

4.2 Challenges and Considerations

- **High Computational Cost:** Despite Top-K$_{\text{top}} sparsity, thenumeroussophisticatedcomponents(4experts, r$ Mamba$)willresultinahigherparametercountandincreasedFLOF$ Mamba$requirescarefulinitialization, robustloadbalancing, andp$
- **Hyperparameter Management:** While automating some choices, new hyperparameters are introduced (e.g., K_{top}, number of harmonics K', number of splines M, anchor points P, mixture components N_M, wavelet parameters N_W). Extensive tuning will still be required.
- **Implementation Complexity:** The development of K-MAMOTE is highly demanding, requiring strong expertise in deep learning frameworks and potentially custom CUDA kernels for optimal performance (e.g., for MatrixKAN-like parallelization of B-splines and wavelets).

- **Data Requirements:** As a highly flexible model, K-MAMOTE will likely require large and diverse datasets to fully realize its potential and prevent overfitting.

5 Experimental Plan

To rigorously validate K-MAMOTE's superiority, we will conduct extensive experiments across diverse domains and tasks:

- **Baselines:** Comprehensive comparison against state-of-the-art time encoding methods, including: HCTE, FTE, LeTE (Fourier-based, Spline-based, and Combined LeTE), and DTKN (if an open-source implementation or comparable re-implementation is feasible).
- **Tasks:**
  - **Time Series Forecasting:** Evaluate on standard datasets (e.g., ETT, Weather, Exchange, Electricity) using common backbone models (e.g., Transformer, Pyraformer, TimesNet). Metrics: MAE, MSE.
  - **Dynamic Graph Link Prediction:** Evaluate on real-world datasets (e.g., Wikipedia, Reddit, MOOC, LastFM) using widely adopted baselines (e.g., TGAT, TGN, TCL, DyGFormer). Metrics: AP, AUC-ROC.
  - **Real-World Application:** Demonstrate effectiveness in a practical scenario such as financial risk control, similar to LeTE's evaluation.
- **Ablation Studies:**
  - Impact of different expert combinations within K-MOTE (e.g., Fourier+Spline vs. Fourier+Kernel vs. Spline+Kernel vs. Wavelet vs. all four).
  - Effectiveness of the dynamic MoE router vs. fixed weighting or single experts.
  - Contribution of the Continuous-Mamba block vs. simpler sequential aggregation.
  - Sensitivity to K_{top} (number of activated experts) and regularization strengths.
- **Qualitative Analysis:** Visualize learned functions for each expert and analyze router activations for different types of temporal inputs, demonstrating interpretability and adaptive behavior. Analyze the learned dynamics within the Continuous-Mamba block.

6 Conclusion

K-MAMOTE represents a principled and innovative advancement in time encoding for machine learning. By dynamically adapting the *type* of underlying basis function through an MoE of specialized experts, and integrating these rich embeddings into a continuous-time sequential memory, K-MAMOTE offers unprecedented flexibility and precision in modeling complex and irregular temporal dynamics. This approach promises to yield superior performance across a wide range of applications, streamline model development by reducing manual tuning, and provide deeper insights into the nature of learned spatio-temporal representations.

References