TIME DELAY SYSTEM FOR SEQUENCE MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing sequence models share a critical limitation: they lack an explicit, interpretable mechanism for content-aware temporal recall. LSTMs compress everything into a single hidden state, thereby losing information about specific past events. Transformers attend to all tokens indiscriminately at quadratic cost, while state-space models like Mamba compress history into a fixed-dimensional state without explicit content-addressable memory. To address this gap, we introduce a novel framework grounded in nonlinear time-delay systems, in which the effective delay is dynamically estimated based on the semantic similarity between current and historical inputs. Our approach features three key innovations: (i) a differentiable semantic lag estimator that computes content-aware soft attention weights over history, producing a weighted historical context and similarity score; (ii) a diagonal state space with closed-form cumulative sum solution, achieving O(1) per-step complexity with no sequential loops; and (iii) an interpretable fuzzy semantic modulation mechanism that learns context-dependent rules for balancing instantaneous processing versus historical recall, where rule centers correspond to prototypical semantic contexts. Experimental results on text classification and generation outperform state-of-the-art methods.

1 INTRODUCTION

Existing sequence modeling architectures, such as Long Short-Term Memory (LSTM) (Sherstinsky, 2020), transformers, and Mamba (Gu & Dao, 2023), although achieving promising performance, share a limitation, that is, a lack of an explicit and interpretable mechanism for temporal recall. This limitation motivates our research on sequence modeling using nonlinear time-delay systems (TDS). In real-world sequential modeling problems, the relevance of historical information depends not only on temporal distance but also on semantic similarity between current and past inputs. Therefore, we propose a semantic delay-driven sequence modeling framework that explicitly estimates adaptive temporal recall based on semantic relevance. Instead of propagating through a single hidden state, as in recurrent neural networks (RNNs), or attending to all tokens, as in transformers, our proposed model dynamically determines which historical states influence the current computation more via differentiable semantic delay estimation and interpretable fuzzy semantic modulation.

Our approach presents three key contributions:

- First, we propose a differentiable semantic lag estimator that computes contextual similarity scores over historical states and produces weighted semantic historical context through soft attention delay aggregation. This provides explicit semantic temporal recall and interpretable delay estimation.
- Second, we design a hidden dynamics using a diagonal state-space structure with a closed-form cumulative sum solution, enabling fully parallelizable O(1) complexity per step without sequential recurrence or quadratic attention computation.
- Third, we introduce an interpretable fuzzy semantic modulation mechanism that adaptively balances current input processing and delayed historical recall using learnable fuzzy rules, providing interpretable semantic control.

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.

2 Related Work

Sequence modeling for NLP enables models to process and generate text by capturing temporal dependencies. Architectures such as recurrent neural networks and state-space models have been proposed to improve the modeling of temporal dependencies (Sherstinsky, 2020).

Recurrent Neural Networks and LSTM. Early sequence modeling approaches primarily relied on RNNs, which operate via a hidden state that evolves over time. However, they suffer from vanishing and exploding gradient problems, limiting their ability to capture long-range dependencies (Bengio et al., 1994). To address this, the LSTM (Hochreiter & Schmidhuber, 1997) incorporates gating mechanisms to regulate information flow, improving its ability to model long-term dependencies. However, a core limitation of LSTM is that it captures temporal differences implicitly, without an explicit representation of how past information is delayed or propagated over time, as all time steps are compressed into the hidden state (Pascanu et al., 2013), making it difficult to interpret in human terms. This lack of explicit temporal semantics limits its ability to model human-like reasoning over the relationship between past contextual information and the present. Therefore, while LSTM can retain relevant information for longer sequences, it does not provide an interpretable mechanism for understanding how certain past inputs influence the current state.

State Space Models and Efficient Sequence Modeling. State space models (SSMs) have recently become a powerful framework for sequence modeling (Rangapuram et al., 2018; Smith et al., 2023; Zhou et al., 2023). The classic state-space model is formulated as a latent state that evolves through structured linear transitions, enabling a continuous-time interpretation of sequences (Kalman, 1960; Kailath, 1980). In recent years, the Structured State Space sequence model (S4) was proposed by Gu et al. (2021), which leveraged HiPPO-based projections to improve the model's ability to process longer-range sequences while remaining computationally efficient. Later, the Mamba architecture proposed by Gu and Dao (2023) introduced selective state space models, where input-dependent gating facilitates state transitions, enabling linear-time complexity while maintaining strong performance across long sequences. Despite these advancements, the SSM-based models primarily achieved computational and structural improvements rather than conceptual reasoning. Specifically, the content representation of structured state transitions remains largely obscured. The learned transition matrices and input-dependent parameters (Gu et al., 2021) do not explicitly encode semantically meaningful notions such as temporal delay, lag structure, or the relevance of past information, leaving the temporal dynamics still lacking interpretability.

Time-Delay Modeling in Dynamical Systems. Time-delay systems have been studied in control theory and dynamical systems to model temporal dependencies with explicit lags (Insperger & Stépán, 2011; Hale & Lunel, 2013). In delay differential equations (DDEs), the system's evolution depends on past states with a defined time delay, enabling explicit modeling of the relationship between temporal memories independent of position in time (Smith, 2011; Waibel et al., 2013; Snyder et al., 2015). Recent work continues to investigate the delay structures in complex dynamical systems. Peddinti et al. (Peddinti et al., 2015) first introduced the Time Delay Neural Network (TDNN) for efficiently modeling long-range temporal dependencies in speech recognition while avoiding the high training cost of recurrent models. Huang et al. (2019) proposed deeper TDNNs for acoustic modeling by increasing the depth of temporal context layers. Ji et al. (2024) further improved TDNNs by introducing trainable and continuous time delays as learnable parameters, allowing the network to directly identify both the magnitude of delays and the underlying nonlinear dynamics from data. However, these approaches remain limited in the context of NLP. Existing TDNN-based models are primarily evaluated on low-dimensional time-series data and have not been widely scaled to high-dimensional sequence modeling tasks such as language understanding or generation. Additionally, traditional time delay networks does not explicitly incorporate semantic meanings to moderate temporal dynamics (Jatnika et al., 2019; Sivakumar et al., 2020), revealing the insufficient semantic awareness and interpretability of recent NLP model designs.

From the review, we can see that current sequence modeling approaches lack an explicit, interpretable mechanism for modeling how semantic information propagates over time with variable delay. RNNs and LSTMs encode temporal dependencies implicitly through hidden states and gating mechanisms. State-space models improve efficiency but remain black boxes. Interpretability methods are largely post-hoc and do not reflect internal temporal processes. Time-delay systems and fuzzy logic offer interpretability but are not well-integrated with modern semantic representations in NLP. This gap

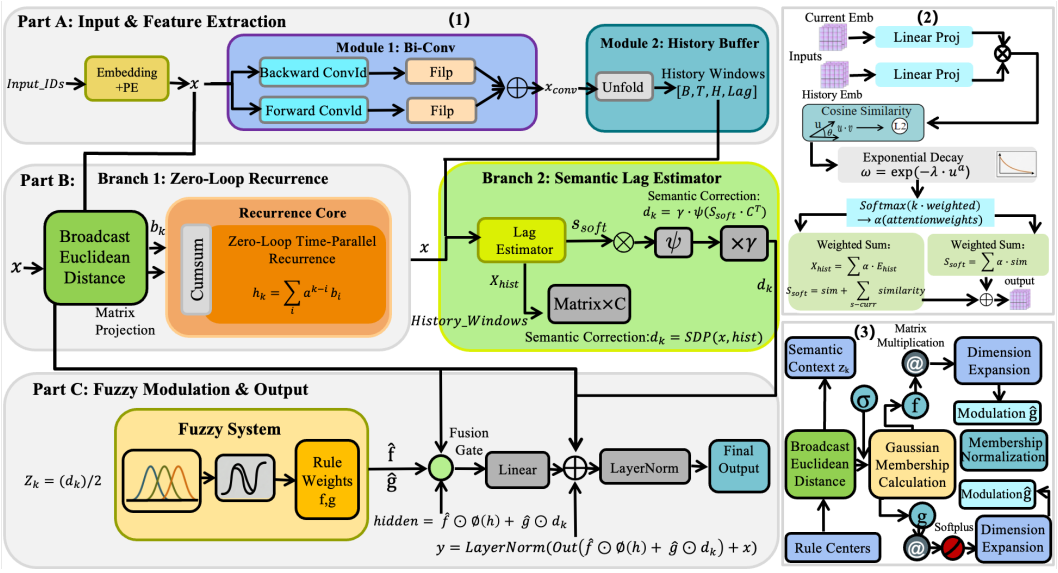

Figure 1: (1) is the overall architecture of our proposed TDS model. (2) is the semantic lag estimator, and (3) is the fuzzy system.

motivates the exploration of new approaches that can integrate semantic understanding, temporal delay, and interpretability. Our proposed semantic delay-driven fuzzy state update model addresses these limitations by integrating semantic delay estimation with fuzzy logic-based control, enabling both effective and interpretable sequence modeling.

3 METHODS

Motivation. Existing sequence models struggle with a fundamental problem: they cannot explicitly decide when to look back in history based on semantic relevance. LSTMs compress everything into a single hidden state, losing specific past events. Transformers look at everything but at quadratic cost—inefficient and unfocused. Mamba improves efficiency but still lacks an interpretable mechanism for content-aware recall. Our time delay system directly addresses this by introducing an explicit semantic lag estimator that answers: "Which past token is most relevant to what I'm seeing now?" This allows the model to dynamically retrieve and integrate specific historical information—much like how humans recall relevant memories when triggered by current input. Combined with interpretable fuzzy rules that learn when to trust the present versus the past, our approach achieves both the expressiveness of attention and the efficiency of linear recurrence, while offering a clear, mathematical foundation grounded in delay differential equations.

3.1 PROBLEM DEFINITION

In conventional recurrent or state-space language models, the evolution of the current hidden state $\boldsymbol{s}(t)$, (where t is the current time) is assumed to depend only on the current input $\boldsymbol{x}(t)$ and the previous state $\boldsymbol{s}(t-1)$. However, natural language often exhibits **semantic delay** $\tau(t)$: words or phrases can influence the meaning of future tokens after a variable time lag. To capture this, we introduce a **nonlinear time-delay dynamical system** as the foundation for sequence modeling.

3.2 MATHEMATICAL FOUNDATION

Formally, let $\boldsymbol{s}(t) \in \mathbb{R}^d$ denote the current state (d is the dimension of the state) and $\boldsymbol{x}(t) \in \mathbb{R}^m$ the current input ($e.g.$, token embedding, m is the dimension of the input). The system evolves according to a nonlinear time-delay differential equation:

$$\dot{\boldsymbol{s}}(t) = \boldsymbol{f}\big(\boldsymbol{s}(t), \boldsymbol{s}(t-\tau(t)), \boldsymbol{x}(t)\big), \tag{1}$$

where $\dot{\boldsymbol{s}}(t)$ is the change rate of the current state $s(t)$; $\tau(t) \in [0, L]$ denotes a time delay that can vary dynamically with the semantic context (L is the maximum of τ ($\tau_{\max}$)); $\boldsymbol{s}(t-\tau(t))$ is delayed past state; and $\boldsymbol{f}$ is a nonlinear function that can simultaneously process the current state, delayed past state and current input.

3.3 Functional Separation

According to the decomposition theory of multivariate functions, any smooth multivariate vector function $\boldsymbol{f}$ can be decomposed at the reference state point $\boldsymbol{s}_0$ as following,

$$\boldsymbol{f}\big(\boldsymbol{s}, \boldsymbol{s}_\tau, \boldsymbol{x}\big) = \boldsymbol{f}\big(\boldsymbol{s}, \boldsymbol{s}_0, \boldsymbol{x}\big) + [\boldsymbol{f}\big(\boldsymbol{s}, \boldsymbol{s}_\tau, \boldsymbol{x}\big) - \boldsymbol{f}\big(\boldsymbol{s}, \boldsymbol{s}_0, \boldsymbol{x}\big)] \tag{2}$$

where $\boldsymbol{s}_\tau = \boldsymbol{s}(t-\tau(t))$. We can define two componentfunctions based on the above split:

- Instantaneous Function: $F(s, x) = \boldsymbol{f}\big(\boldsymbol{s}, \boldsymbol{s}_0, \boldsymbol{x}\big)$, which captures the system's behavior when the delayed state is at its reference value.
- Delay Function: $G\big(\boldsymbol{s}, \boldsymbol{s}_\tau, \boldsymbol{x}\big) = \boldsymbol{f}\big(\boldsymbol{s}, \boldsymbol{s}_\tau, \boldsymbol{x}\big) - \boldsymbol{f}\big(\boldsymbol{s}, \boldsymbol{s}_0, \boldsymbol{x}\big)$, which captures the deviation caused by the delayed state being different from the reference.

Substituting these two definitions back to Eq. (2), we can get

$$\dot{\boldsymbol{s}}(t) = F\big(\boldsymbol{s}(t), \boldsymbol{x}(t)\big) + G\big(\boldsymbol{s}(t), \boldsymbol{s}(t-\tau(t)), \boldsymbol{x}(t)\big). \tag{3}$$

Eq. (3) is still complex since G depends on both the current state $\boldsymbol{s}(t)$ and the delayed state $\boldsymbol{s}(t-\tau(t))$. For a more practical and interpretable model, we introduce a physically-motivated simplification: the primary function of the delay is to inject information from the past history, thus, we approximate the delay function as being independent of the current state as follows:

$$G\big(\boldsymbol{s}(t), \boldsymbol{s}(t-\tau(t)), \boldsymbol{x}(t)\big) \approx \tilde{G}(\boldsymbol{s}(t-\tau(t)), \boldsymbol{x}(t-\tau(t))) \tag{4}$$

Hence, we can rewrite Eq. (3) as:

$$\dot{\boldsymbol{s}}(t) = F\big(\boldsymbol{s}(t), \boldsymbol{x}(t)\big) + G(\boldsymbol{s}(t-\tau(t)), \boldsymbol{x}(t-\tau(t))) \tag{5}$$

where the first term $F(\cdot)$ captures the instantaneous state evolution driven by the current input and the second term $G(\cdot)$ introduces delayed feedback that depends on past states and inputs. We drop the tilde on G for notation simplicity. Eq. (5) is the fundamental definition of a continuous nonlinear time-delay system. It subsumes the standard state-space model when $\tau(t) = 0$ and $G \equiv 0$.

3.4 Structure Parameterization

To obtain a learnable form for neural network modeling, we introduce structured linear mappings and nonlinear activations as follows:

$$F(\mathbf{s}(t), \mathbf{x}(t)) = \phi\big(\mathbf{A}\mathbf{s}(t) + \mathbf{B}\mathbf{x}(t)\big), \tag{6}$$

$$G(\mathbf{s}(t-\tau(t)), \mathbf{x}(t-\tau(t))) = \gamma\, \psi\big(\hat{s}(t)\, \mathbf{C}\, \bar{\mathbf{x}}(t-\tau(t))\big), \tag{7}$$

where $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{R}^{d\times d}$ are learnable matrices, $\phi(\cdot)$ and $\psi(\cdot)$ are nonlinear activation functions (e.g., SiLU, ReLU), γ is a scalar weight controlling the influence of the delayed term, and $\hat{s}(t) \in [0, 1]$ is the semantic similarity between the current and delayed embeddings. Substituting Eq. (6) and Eq. (7) into Eq. (5) yields:

$$\dot{\mathbf{s}}(t) = \phi\big(\mathbf{A}\mathbf{s}(t) + \mathbf{B}\mathbf{x}(t)\big) + \gamma\, \psi\big(\hat{s}(t)\, \mathbf{C}\, \mathbf{x}(t-\tau(t))\big). \tag{8}$$

This equation describes a nonlinear dynamical system with both instantaneous and delayed semantic interactions.

3.5 Discretization

For sequential data (*e.g.*, text), the system evolves in discrete time steps. Let the time discretization be $t_k = k\Delta t$, and approximate the time derivative in Eq. (8) using the first-order forward difference:

$$\dot{\mathbf{s}}(t_k) \approx \frac{\mathbf{s}_k - \mathbf{s}_{k-1}}{\Delta t}. \tag{9}$$

Substituting this into Eq. (8) gives:

$$\frac{\mathbf{s}_k - \mathbf{s}_{k-1}}{\Delta t} = \phi\big(\mathbf{A}\mathbf{s}_{k-1} + \mathbf{B}\mathbf{x}_k\big) + \gamma\,\psi\big(\hat{s}_k\,\mathbf{C}\,\mathbf{x}_{k-\tau_k}\big). \tag{10}$$

Rearranging terms leads to the discrete-time update:

$$\mathbf{s}_k = \mathbf{s}_{k-1} + \Delta t\left[\phi\big(\mathbf{A}\mathbf{s}_{k-1} + \mathbf{B}\mathbf{x}_k\big) + \gamma\,\psi\big(\hat{s}_k\,\mathbf{C}\,\mathbf{x}_{k-\tau_k}\big)\right]. \tag{11}$$

When the sampling step is normalized to $\Delta t = 1$, the additive integration term can be absorbed into the nonlinear mappings, yielding a simplified discrete nonlinear delay model:

$$\mathbf{s}_k = \phi\big(\mathbf{A}\mathbf{s}_{k-1} + \mathbf{B}\mathbf{x}_k\big) + \gamma\,\psi\big(\hat{s}_k\,\mathbf{C}\,\mathbf{x}_{k-\tau_k}\big), \tag{12}$$

which is the core discrete-time realization of our nonlinear time-delay system.

3.6 Semantic Lag Estimator

To calculate the second delay term $\gamma\psi\big(\hat{s}_k\,\mathbf{C}\,\mathbf{x}_{k-\tau_k}\big)$, we first need to consider the semantic delay $\tau(t)$, which is determined by the similarity between the current input and its past history. We define a semantic lag estimator that computes the similarity $\mathcal{S}$ between the current embedding $\boldsymbol{x}_t$ and each historical embedding $\boldsymbol{x}_{t-k}$ within the window, where each lag $k = 1, \ldots, L$ and L is the maximum number of semantic delay τ. Here we define $\mathcal{S}$, the cosine similarity as follows:

$$\mathrm{sim}_k = \frac{\langle \boldsymbol{x}_t, \boldsymbol{x}_{t-k} \rangle}{\|\boldsymbol{x}_t\|_2\,\|\boldsymbol{x}_{t-k}\|_2}, \tag{13}$$

where $||\cdot||_2$ is the L2 norm. To suppress far-away history contexts, each similarity value is weighted by a temporal decay function $w_k = \exp(-\lambda k^\alpha)$,, where $\lambda > 0$ and $\alpha > 0$ are hyperparameters controlling the rate of exponential decay. The combined semantic activation over the past history window $\mathcal{W}_t = \{\boldsymbol{x}_{t-k}\}_{k=1}^{L}$ is defined as $\tilde{s}_{t,k} = w_k\,\mathrm{sim}_k$,, and the normalized semantic confidence is

$$\hat{s}_t = \sigma\left(\sum_{k=1}^{L} \tilde{s}_{t,k}\right), \tag{14}$$

where $\sigma(\cdot)$ denotes the sigmoid function.

Hard selection (interpretable). The effective discrete delay is selected as the smallest lag exceeding an input-dependent threshold:

$$\tau_t = \min\left\{k \in \{1, \ldots, L\} : \tilde{s}_{t,k} \geq \theta \cdot \sigma(\|\mathbf{x}_t\|)\right\}, \tag{15}$$

with corresponding similarity score $\hat{s}_t = \tilde{s}_{t,\tau_t}$. If no lag satisfies the condition, we set $\tau_t = 0$ and $\hat{s}_t = 0$.

Soft selection (differentiable). To enable end-to-end training, we introduce a soft attention mechanism over lags:

$$\alpha_{t,k} := \frac{\exp\left(\kappa \tilde{s}_{t,k}\right)}{\sum_{j=1}^{L} \exp\left(\kappa \tilde{s}_{t,j}\right)} \qquad (\kappa > 0), \quad \bar{x}_t^{\mathrm{hist}} \quad := \sum_{k=1}^{L} \alpha_{t,k}\, x_{t-k}, \quad \hat{s}_t^{\mathrm{soft}} := \sum_{k=1}^{L} \alpha_{t,k}\, \tilde{s}_{t,k}. \tag{16}$$

Then, we have:

$$\tau_t^{\mathrm{soft}} = \sum_{k=1}^{L} k \cdot \alpha_{t,k}. \tag{17}$$

Substituting these expressions into Eq. (12) gives the continuous and differentiable soft-delay update:

$$\mathbf{s}_k = \underbrace{\phi\big(\mathbf{A}\mathbf{s}_{k-1} + \mathbf{B}\mathbf{x}_k\big)}_{\text{Base Update}} + \underbrace{\gamma\,\psi\big(\hat{s}_k^{\mathrm{soft}}\,\mathbf{C}\,\bar{\mathbf{x}}_k^{\mathrm{hist}}\big)}_{\text{Delay Correction}}, \tag{18}$$

with $s_t = \bar{s}_t + \Delta_t^{\mathrm{soft}}$. In the limit $\kappa \to \infty$, the soft quantities concentrate on the minimal lag $\tilde{s}_{t,k}$, and τ_t^{soft} recovers the hard selection τ_t, where $\mathbf{s}_k \in \mathbb{R}^d$ is the semantic state at time k, $\mathbf{x}_k \in \mathbb{R}^d$ is the current token embedding, $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{R}^{d\times d}$ are learnable matrices, ϕ, ψ are nonlinear activation functions (e.g., SiLU, tanh), $\gamma \in \mathbb{R}$ is a global scaling parameter, $\hat{s}_k^{\mathrm{soft}} \in \mathbb{R}$ is the soft similarity score from the Semantic Lag Estimator, and $\bar{\mathbf{x}}_k^{\mathrm{hist}} \in \mathbb{R}^d$ is the soft-weighted historical context.

Algorithm 1 Soft Semantic Lag Estimator

Require: Current embeddings $\mathbf{X}_{\text{curr}} \in \mathbb{R}^{B\times H}$ or $\mathbb{R}^{B\times T\times H}$, History windows $\mathbf{H} \in \mathbb{R}^{B\times L\times H}$ or $\mathbb{R}^{B\times T\times L\times H}$, Parameters: projection $\mathbf{W}_{\text{proj}}$, decay λ, α, temperature κ

Ensure: Attention weights $\boldsymbol{\alpha}$, weighted history $\bar{\mathbf{x}}^{\text{hist}}$, soft score $\hat{s}^{\text{soft}}$

Initialize: $w_u = \exp(-\lambda u^{\alpha}),\ u = 1, \ldots, L$ ▷ Fixed decay weights

$\mathbf{X}_{\text{curr}}^{\text{proj}} = \mathbf{W}_{\text{proj}}\mathbf{X}_{\text{curr}}$ ▷ Project to similarity space

$\mathbf{H}^{\text{proj}} = \mathbf{W}_{\text{proj}}\mathbf{H}$

$$\text{sim} = \frac{\mathbf{X}_{\text{curr}}^{\text{proj}} \cdot {\mathbf{H}^{\text{proj}}}^{\top}}{\|\mathbf{X}_{\text{curr}}^{\text{proj}}\| \cdot \|\mathbf{H}^{\text{proj}}\| + \varepsilon}$$ ▷ Cosine similarity

$\tilde{s}_{k,u} = \text{sim}_{k,u} \cdot w_u$ ▷ Apply temporal decay

$$\alpha_{k,u} = \frac{\exp(\kappa \tilde{s}_{k,u})}{\sum_j \exp(\kappa \tilde{s}_{k,j})}$$ ▷ Softmax over history

$\bar{\mathbf{x}}^{\text{hist}} = \sum_u \alpha_{k,u}\mathbf{x}_{k-u}$ ▷ Weighted historical context

$\hat{s}^{\text{soft}} = \sum_u \alpha_{k,u}\tilde{s}_{k,u}$ ▷ Soft similarity score

return $\boldsymbol{\alpha},\ \bar{\mathbf{x}}^{\text{hist}},\ \hat{s}^{\text{soft}}$

3.7 Fuzzy Semantic System

We define the delay correction term in the core equation Eq. (18) as:

$$\mathbf{d}_k = \gamma\, \psi\big(\hat{s}_k^{\text{soft}}\, \mathbf{C}\, \bar{\mathbf{x}}_k^{\text{hist}}\big), \tag{19}$$

and the equation simplifies to:

$$\mathbf{s}_k = \phi\big(\mathbf{A}\mathbf{s}_{k-1} + \mathbf{B}\mathbf{x}_k\big) + \mathbf{d}_k. \tag{20}$$

The core equation treats both terms with equal weight (coefficient 1 for each term). However, different semantic contexts require different balances. To address this, we introduce context-dependent modulation:

$$\mathbf{s}_k = f(\mathbf{z}_k) \cdot \phi\big(\mathbf{A}\mathbf{s}_{k-1} + \mathbf{B}\mathbf{x}_k\big) + g(\mathbf{z}_k) \cdot \mathbf{d}_k, \tag{21}$$

where $\mathbf{z}_k \in \mathbb{R}^d$ is the semantic context vector, and $f, g : \mathbb{R}^d \to \mathbb{R}$ are modulation functions.

Semantic Context Definition. The semantic context captures both the current and recently recalled information:

$$\mathbf{z}_k = \frac{\mathbf{s}_{k-1} + \mathbf{d}_k}{2}. \tag{22}$$

Fuzzy Rule-Based Parameterization. We employ a zero-order Takagi-Sugeno-Kang (TSK) fuzzy system with R rules. Each rule r has a prototypical semantic context $\mathbf{c}_r \in \mathbb{R}^d$, a consequent $f_r \in \mathbb{R}$ for base update modulation, and a consequent $g_r \in \mathbb{R}$ for delay correction modulation.

Membership Function. The unnormalized membership is given by a Gaussian radial basis function:

$$\tilde{\mu}_r(\mathbf{z}) = \exp\left(-\frac{\|\mathbf{z} - \mathbf{c}_r\|^2}{2\sigma^2}\right), \tag{23}$$

where $\sigma > 0$ is the width parameter. The normalized memberships are:

$$\mu_r(\mathbf{z}) = \frac{\tilde{\mu}_r(\mathbf{z})}{\sum_{j=1}^{R} \tilde{\mu}_j(\mathbf{z}) + \varepsilon}, \qquad \varepsilon = 10^{-8}, \tag{24}$$

which satisfy: $0 \le \mu_r(\mathbf{z}) \le 1, \quad \sum_{r=1}^{R} \mu_r(\mathbf{z}) = 1.$

Algorithm 2 Fuzzy Semantic System

Require: Semantic context $\mathbf{z} \in \mathbb{R}^{B\times H}$ or $\mathbb{R}^{B\times T\times H}$, Parameters: rule centers $\{\mathbf{c}_r\}_{r=1}^{R} \subset \mathbb{R}^{H}$, rule weights $\{f_r\}_{r=1}^{R}, \{g_r\}_{r=1}^{R} \subset \mathbb{R}$, width $\sigma > 0$

Ensure: Modulation vectors $\hat{\mathbf{f}}, \hat{\mathbf{g}}$ (same shape as $\mathbf{z}$)

1: $\mu_r(\mathbf{z}) = \dfrac{\exp\left(-\dfrac{\|\mathbf{z}-\mathbf{c}_r\|_2^2}{2\sigma^2}\right)}{\sum_{j=1}^{R}\exp\left(-\dfrac{\|\mathbf{z}-\mathbf{c}_j\|_2^2}{2\sigma^2}\right)+\varepsilon}$ ▷ Normalized membership

2: $\hat{f}_{\text{scalar}} = \sum_{r=1}^{R}\mu_r(\mathbf{z})\cdot f_r$ ▷ Weighted average for f

3: $\hat{g}_{\text{scalar}} = \text{softplus}\Big(\sum_{r=1}^{R}\mu_r(\mathbf{z})\cdot g_r\Big)$ ▷ Weighted average for g (non-negative)

4: $\hat{\mathbf{f}} = \hat{f}_{\text{scalar}}\cdot\mathbf{1}_H, \quad \hat{\mathbf{g}} = \hat{g}_{\text{scalar}}\cdot\mathbf{1}_H$ ▷ Expand to hidden dimension

5: **return** $\hat{\mathbf{f}}, \hat{\mathbf{g}}$

subsubsectionDefuzzifications. The modulation outputs are weighted averages of the rule consequents:

$$f(\mathbf{z}) = \sum_{r=1}^{R}\mu_r(\mathbf{z})\cdot f_r, \qquad g(\mathbf{z}) = \sum_{r=1}^{R}\mu_r(\mathbf{z})\cdot g_r. \tag{25}$$

Positivity Constraint. The delay modulation is constrained to be positive via the softplus function:

$$\tilde{g}(\mathbf{z}) = \text{softplus}\big(g(\mathbf{z})\big) = \ln\Big(1+e^{g(\mathbf{z})}\Big) > 0. \tag{26}$$

Complete Fuzzy-Modulated State Update. Substituting all components, the complete system is:

$$f_k = \sum_{r=1}^{R}\mu_r(\mathbf{z}_k)\,f_r, \quad g_k = \ln\left(1+\exp\left(\sum_{r=1}^{R}\mu_r(\mathbf{z}_k)\,g_r\right)\right). \tag{27}$$

We can get the final updated core function as follows:

$$\mathbf{s}_k = f_k\cdot\phi\big(\mathbf{A}\mathbf{s}_{k-1}+\mathbf{B}\mathbf{x}_k\big) + g_k\cdot\mathbf{d}_k. \tag{28}$$

3.8 Time Delay System

In Eq. (28), we need to recursively get s_k using the last state s_{k-1}. To achieve fully vectorized computation without sequential loops, we restrict the state transition matrix $\mathbf{A}$ to be diagonal. Let $\mathbf{a} = \text{diag}(\mathbf{A}) \in \mathbb{R}^d$ denote the diagonal elements. For $\mathbf{A}\mathbf{s}_{k-1} + \mathbf{B}\mathbf{x}_k$, we apply the linear recurrence with diagonal state transition. The latent state $\mathbf{h}_k \in \mathbb{R}^d$ evolves according to the element-wise recurrence:

$$\mathbf{h}_k = \mathbf{a}\odot\mathbf{h}_{k-1} + \mathbf{b}_k, \quad \mathbf{h}_0 = \mathbf{0}, \tag{29}$$

where $\odot$ denotes element-wise (Hadamard) multiplication, and $\mathbf{b}_k = \mathbf{B}\mathbf{x}_k \in \mathbb{R}^d$ is the projected input.

Closed-Form Solution via Cumulative Sum. Unrolling the recurrence yields the closed-form expression:

$$\mathbf{h}_k = \sum_{i=1}^{k}\Big(\mathbf{a}^{\odot(k-i)}\odot\mathbf{b}_i\Big), \tag{30}$$

where $\mathbf{a}^{\odot p}$ denotes element-wise exponentiation: $(\mathbf{a}^{\odot p})_j = (a_j)^p$. This can be computed efficiently using cumsum operation:

$$\mathbf{h}_k = \mathbf{a}^{\odot k} \odot \mathrm{cumsum}_{i=1}^{k} \left(\mathbf{a}^{\odot -i} \odot \mathbf{b}_i \right), \tag{31}$$

where $\mathbf{a}^{\odot -i}$ denotes element-wise reciprocal: $(\mathbf{a}^{\odot -i})_j = (a_j)^{-i}$.

Final State Update. The semantic state $\mathbf{s}_k$ is obtained by combining the transformed latent state and delay correction with fuzzy modulation:

$$\mathbf{s}_k = \hat{\mathbf{f}}_k \odot \phi(\mathbf{h}_k) + \hat{\mathbf{g}}_k \odot \mathbf{d}_k, \tag{32}$$

where ϕ is a nonlinear activation function (e.g., SiLU).

Algorithm 3 Semantic Delay Processor (Full version is shown in Alg. 6.)

Require: Input tokens $I \in \mathbb{N}^{B \times T}$, parameters Θ
Ensure: Final state $\mathbf{s}_{\text{final}} \in \mathbb{R}^{B \times H}$, average state $\bar{\mathbf{s}} \in \mathbb{R}^{B \times H}$

$\mathbf{X} = \text{Dropout}(\text{Embed}(I) + \text{PosEnc}(1:T))$ ▷ Embedding
$\mathbf{X}_{\text{conv}} = \text{GELU}(\text{Conv}_{\text{bi}}(\mathbf{X}))$ ▷ Local context
$(\boldsymbol{\alpha},\ \bar{\mathbf{X}}^{\text{hist}},\ \hat{s}) = \text{LagEstimator}(\mathbf{X},\ \text{History}(\mathbf{X}))$ ▷ Algorithm 1
$\mathbf{b} = \mathbf{B}\mathbf{X} \in \mathbb{R}^{B \times T \times H}$ ▷ Projected input
$\mathbf{d} = \gamma \tanh(\hat{s} \odot \mathbf{C}\bar{\mathbf{X}}^{\text{hist}})$ ▷ Delay correction
$\mathbf{a} = \text{diag}(\mathbf{A})$, $\mathbf{a}_t^{\text{pow}} = \mathbf{a}^{\odot t}$
$\mathbf{h} = \mathbf{a}^{\odot t} \odot \text{CumSum}(\mathbf{b} \odot \mathbf{a}^{\odot -t})$ ▷ Parallel scan
$(\hat{\mathbf{f}},\ \hat{\mathbf{g}}) = \text{FuzzySystem}(\mathbf{d}/2)$ ▷ Algorithm 2
$\mathbf{s} = \hat{\mathbf{f}} \odot \text{SiLU}(\mathbf{h}) + \hat{\mathbf{g}} \odot \mathbf{d}$ ▷ State update
$\mathbf{s} = \text{LayerNorm}(\mathbf{W}_{\text{out}}\mathbf{s} + \mathbf{X})$ ▷ Residual + output
return $\mathbf{s}[:, -1, :],\ \text{Mean}(\mathbf{s}, \text{dim} = 1)$

4 EXPERIMENTS

4.1 DATASETS AND IMPLEMENTATIONS

In this paper, we test our TDS model on the **IMDB** (Maas et al., 2011) dataset for text classification and the **CNN / DailyMail** (Nallapati et al., 2016) dataset for text generation. IMDB has 25,000 training and 25,000 test samples, while CNN / DailyMail contains over 300,000 unique news samples. We use the full dataset for text classification training and testing, while for the generation task, we use 80,000 samples for training and 2,000 each for validation and testing. The detailed algorithms for text classification and generation are shown in Alg. 4 and Alg. 5 in the Appendix.

We run our experiments using PyTorch on an Nvidia H100 80G GPU and use the Adam optimizer. We set the learning rate to 3e-4 for text classification and 5e-5 for text generation; the batch size is 32; the number of iterations is 50 for classification and 100 for generation. We set the dropout rate to 0.3 to improve model generalization and avoid overfitting during training. Our proposed TDS model employs a hidden dimension of 256, a maximum semantic delay length of 30, and 6 fuzzy inference rules for adaptive state updating. [1].

4.2 RESULTS

For the sequence modeling experiments, we also compare the performance of our TDS model with other state-of-the-art models. As shown in Tables 1 and 2, our proposed TDS model outperforms state-of-the-art methods on text classification and generation tasks, achieving higher accuracy and ROUGE scores while maintaining a relatively small parameter count and efficient training time. However, due to the limited number of benchmark experiments, the model's broader generalizability remains untested. Meanwhile, the TDS model is also more sensitive to hyperparameter tuning.

[1] Source code will be available upon acceptance.

Table 1: Number of parameters and accuracy of the IMDB dataset.

Model	Parameters(M)	Accuracy	Total Training Time(s)
LSTM (Qaisar, 2020)	2.8	89.90	2300
Mamba (Gu & Dao, 2023)	3.2	83.00	2300
RoBERTa-base (Abidin et al., 2025)	125	94.13	-
DistilBERT-base (Abidin et al., 2025)	66	92.13	-
TDS	**2.6**	**95.35**	**2100**

Table 2: Number of parameters and ROUGE score of the CNN/DailyMail dataset.

Model	Parameters(M)	ROUGE-L	Total Training Time(s)
LSTM (Abdulsahib et al., 2022)	26.5	38.45	16558
BERT (Maharditha & Lhaksmana, 2025)	24.30	40.48	18260
Mamba (Do et al., 2025)	14.73	29.60	16725
BART CNN-large (Rani Krishna et al., 2025)	139	24.58	-
T5-base (Rani Krishna et al., 2025)	220	23.54	-
TDS	16.98	42.63	17633

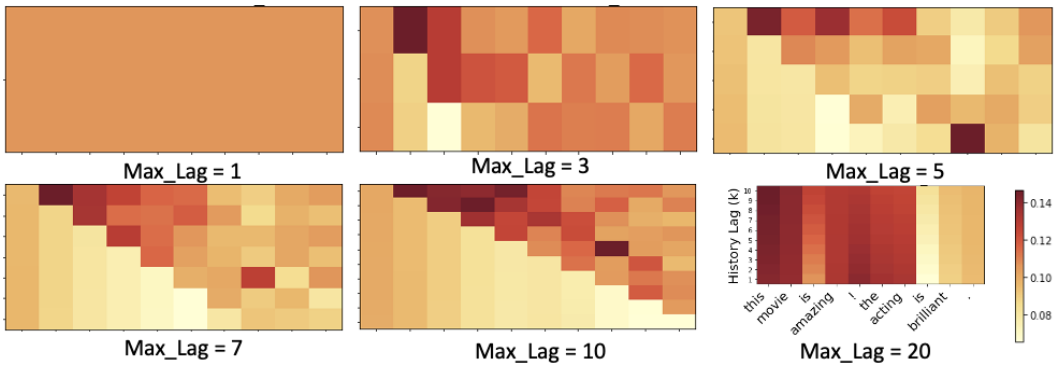

Figure 2: An example of how the maximum number of lag L impacts TDS model performance on the sample “this movie is amazing! the acting is brilliant.".

We also visualize how different numbers of the maximum number of lags affect model performance, as shown in Fig. 2, and Fig. 3 in the Appendix. The X-axis is a sample sentence of ten words: “this movie is amazing ! the acting is brilliant. The y-axis is the maximum number of historical lags. Given an example when Max_Lag = 10, the word “amazing" (position 3) attends to a maximum of the past 10 historical words. The figure shows that “movie" has the biggest impact on the word “amazing". We can see from Fig. 2 that as the number of lags increases, the importance of specific historical words to the current sequence becomes more prominent. While attention values can vary during training with different lags L, the most relevant historical context will be consistently highlighted by the attention mechanism. Fig. 3 shows that increasing lags, although with fluctuation, will increase the model accuracy on text classification.

5 Conclusion

In this paper, we introduced a novel sequence modeling framework based on time-delay systems (TDS), where a differentiable semantic lag estimator dynamically retrieves relevant history via content-aware attention. Our Semantic Delay Processor (SDP) explicitly extracts memories and follows an interpretable context-dependent processing rule. Experiments demonstrate that TDS outperforms RNN, Transformer, and Mamba on sentiment classification and text summarization tasks, while scaling linearly to million-length sequences. The learned fuzzy rule centers correspond to semantic contexts, offering interpretability. Our work explores time-delay systems as an efficient and interpretable mechanism for sequence modeling, which has proven effective for text tasks. It has the potential to be applied to solving more real-life problems, such as image and audio processing and generation, given its efficiency and offering greater interpretability.

AI USE STATEMENT

In preparing this version of the manuscript, we used generative AI tools to assist with conversion to the ICLR 2027 LaTeX format, package and file-path consistency checks, and the identification of compilation issues. We did not use generative AI tools in this formatting process to generate experimental data, conduct statistical analyses, select or fabricate references, or draw scientific conclusions. We reviewed all AI-assisted changes and verified the resulting LaTeX source and compiled document. We take responsibility for the final content of this work, including all text, claims, results, and artifacts.

REFERENCES

Muna Ghazi Abdulsahib et al. Convolutional recurrent neural networks for text lecture summarization. *Iraqi Journal of Computers, Communications, Control and Systems Engineering*, 22(2):3, 2022.

Dodo Zaenal Abidin, Lasmedi Afuan, Afrizal Nehemia Toscany, and Nurhadi Nurhadi. A comprehensive benchmarking pipeline for transformer-based sentiment analysis using cross-validated metrics. *Jurnal Teknik Informatika (Jutif)*, 6(4):1797–1810, 2025.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

Duc Anh Do, Luu Anh Tuan, Wray Buntine, et al. Discrete diffusion language model for efficient text summarization. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 6278–6290, 2025.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

Jack K Hale and Sjoerd M Verduyn Lunel. *Introduction to functional differential equations*, volume 99. Springer Science & Business Media, 2013.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Xiaorong Huang, Weibin Zhang, Xiangmin Xu, Ruixiang Yin, and Dongpeng Chen. Deeper time delay neural networks for effective acoustic modelling. *Journal of Physics: Conference Series*, 1229(1):012076, 2019.

Tamás Insperger and Gábor Stépán. *Semi-discretization for time-delay systems: stability and engineering applications*, volume 178. Springer Science & Business Media, 2011.

Derry Jatnika, Moch Arif Bijaksana, and Arie Ardiyanti Suryani. Word2vec model analysis for semantic similarities in english words. *Procedia Computer Science*, 157:160–167, 2019.

Xunbi A Ji and Gábor Orosz. Trainable delays in time delay neural networks for learning delayed dynamics. *IEEE Transactions on Neural Networks and Learning Systems*, 36(3):5219–5229, 2024.

Thomas Kailath. *Linear systems*, volume 156. Prentice-Hall Englewood Cliffs, NJ, 1980.

Rudolph Emil Kalman. A new approach to linear filtering and prediction problems, 1960.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 142–150, 2011.

I Made Denis Maharditha and Kemas M Lhaksmana. Comparative performance analysis of lstm and transformer models for abstractive text summarization. In *2025 International Conference on Data Science and Its Applications (ICoDSA)*, pp. 213–218. IEEE, 2025.

Ramesh Nallapati, Bowen Zhou, Cicero Dos Santos, Çağlar Gulçehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of the 20th SIGNLL conference on computational natural language learning*, pp. 280–290, 2016.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pp. 1310–1318. Pmlr, 2013.

Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. A time delay neural network architecture for efficient modeling of long temporal contexts. In *Proc. Interspeech 2015*, pp. 3214–3218, 2015.

Saeed Mian Qaisar. Sentiment analysis of imdb movie reviews using long short-term memory. In *2020 2nd International Conference on Computer and Information Sciences (ICCIS)*, pp. 1–4. IEEE, 2020.

Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. *Advances in neural information processing systems*, 31, 2018.

KM Rani Krishna, K Somasundaram, P Arulmozhivarman, Sarah A Immanuel, and ER Rajkumar. Deep learning for text summarization using nlp for automated news digest. *Scientific Reports*, 15 (1):36343, 2025.

Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica d: Nonlinear phenomena*, 404:132306, 2020.

Soubraylu Sivakumar, Lakshmi Sarvani Videla, T Rajesh Kumar, J Nagaraj, Shilpa Itnal, and D Haritha. Review on word2vec word embedding neural net. In *2020 international conference on smart electronics and communication (ICOSEC)*, pp. 282–290. IEEE, 2020.

Hal L Smith. *An introduction to delay differential equations with applications to the life sciences*, volume 57. springer New York, 2011.

Jimmy TH Smith, Andrew Warrington, and Scott W Linderman. Simplified state space layers for sequence modeling, 2023. *URL https://arxiv. org/abs/2208.04933*, 2208:04933, 2023.

David Snyder, Daniel Garcia-Romero, and Daniel Povey. Time delay deep neural network-based universal background models for speaker recognition. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 92–97. IEEE, 2015.

Alexander Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J Lang. Phoneme recognition using time-delay neural networks. In *Backpropagation*, pp. 35–61. Psychology Press, 2013.

Linqi Zhou, Michael Poli, Winnie Xu, Stefano Massaroli, and Stefano Ermon. Deep latent state space models for time-series generation. In *International Conference on Machine Learning*, pp. 42625–42643. PMLR, 2023.

A Appendix

A.1 Application algorithm

The detailed algorithms for text classification and text generation are shown in Alg. 4 and Alg. 5.

Algorithm 4 Text Classification with Semantic Delay Processor

Require: Input $I \in \mathbb{N}^{B\times T}$, classes C
Ensure: Probabilities $\mathbf{p} \in \mathbb{R}^{B\times C}$, predictions $\hat{y} \in \mathbb{N}^{B}$
$\mathbf{X}_{\text{pool}} = \text{Mean}(\text{Emb}(I) + \text{PosEnc})$ ▷ $[B, H]$
$(\mathbf{s}_{\text{final}},\ \bar{\mathbf{s}}) = \text{Processor}(I)$ ▷ Alg. 3
$\mathbf{s} = \text{LayerNorm}(\mathbf{W}_{\text{fuse}}[\mathbf{s}_{\text{final}}; \bar{\mathbf{s}}] + \mathbf{W}_{\text{short}}\mathbf{X}_{\text{pool}})$
$\mathbf{p} = \text{Softmax}(\text{MLP}(\mathbf{s}))$ ▷ MLP : $H \to H/2 \to C$
return $\mathbf{p},\ \arg\max(\mathbf{p})$

Algorithm 5 Text Generation with Semantic Delay Processor

Require: Prefix $I_{\text{p}} \in \mathbb{N}^{B\times T_{\text{p}}}$, length T_{gen}, temp τ, top-k
Ensure: Generated sequences $I_{\text{gen}} \in \mathbb{N}^{B\times(T_{\text{p}}+T_{\text{gen}})}$
$I_{\text{gen}} \leftarrow I_{\text{p}}, \quad \mathbf{s} \leftarrow \text{Processor}(I_{\text{p}})$
for $t = 1$ **to** T_{gen} **do**
  $\mathbf{x} = \text{Emb}(I_{\text{gen}}[:, -1]) + \text{PosEnc}(T_{\text{p}} + t)$
  $\mathbf{b} = \mathbf{B}\mathbf{x}, \quad \mathbf{d} = \gamma \tanh(\hat{s} \odot \mathbf{C}\bar{\mathbf{X}}^{\text{hist}})$
  $\mathbf{h} = \mathbf{a}^{\odot t} \odot \text{CumSum}(\mathbf{b} \odot \mathbf{a}^{\odot -t})$
  $(\mathbf{f}, \mathbf{g}) = \text{Fuzzy}(\mathbf{d}/2), \quad \mathbf{s} = \mathbf{f} \odot \text{SiLU}(\mathbf{h}) + \mathbf{g} \odot \mathbf{d}$
  $\mathbf{z} = \mathbf{W}_{\text{out}}\mathbf{s}, \quad \mathbf{z}[\neg\text{TopK}(\mathbf{z}/\tau, k)] = -\infty$
  $w \sim \text{Categorical}(\text{Softmax}(\mathbf{z}))$
  $I_{\text{gen}} \leftarrow \text{Concat}(I_{\text{gen}}, w)$
end for
return I_{gen}

A.2 Computational Complexity

The full model operations are defined in Eq.(32), which are fully vectorized:

- $\mathbf{a}^{\odot k}$ and $\mathbf{a}^{\odot -i}$ are computed via broadcasted exponentiation
- cumsum is a single GPU-accelerated primitive
- All other operations are element-wise or matrix multiplications

The overall time complexity is $\mathcal{O}(1)$ with respect to sequence length when executed on parallel hardware, as the cumsum operation has $\mathcal{O}(\log T)$ in depth but implemented as a single kernel call.

A.3 Special Cases

$$\text{If } \mathbf{a} = \mathbf{1}: \quad \mathbf{h}_k = \mathrm{cumsum}_{i=1}^{k} \mathbf{b}_i \quad \text{(standard cumulative sum)} \tag{33}$$

$$\text{If } \mathbf{a} \to \mathbf{0}: \quad \mathbf{h}_k \approx \mathbf{b}_k \quad \text{(no memory)} \tag{34}$$

$$\text{If } \gamma = 0: \quad \mathbf{s}_k = f_k \cdot \phi(\mathbf{h}_k) \quad \text{(delay correction disabled)} \tag{35}$$

A.4 Effects of different maximum lag and accuracy

B Theoretical Properties

The nonlinear time-delay system defined in Eq. (8) and its discrete realization in Eq. (12) possess several fundamental theoretical properties that guarantee the well-posed and stable dynamics. We summarize the key results as follows.

Algorithm 6 Semantic Delay Processor (Fully Version)

Require: Input tokens $I \in \mathbb{N}^{B\times T}$ (batch size B, sequence length T),
Vocabulary size V, hidden dimension H, maximum lag L, number of fuzzy rules R,
Hyperparameters: $\kappa, \lambda, \alpha, \sigma, \gamma$, kernel size K
Ensure: Final semantic state $\mathbf{s}_{\text{final}} \in \mathbb{R}^{B\times H}$,
Average semantic state $\bar{\mathbf{s}} \in \mathbb{R}^{B\times H}$

Step 1: Token Embedding and Position Encoding
$\mathbf{X} = \text{Embedding}(I) \in \mathbb{R}^{B\times T\times H}$ ▷ Lookup embedding table
$\mathbf{X} = \mathbf{X} + \text{PosEnc}(1:T) \in \mathbb{R}^{B\times T\times H}$ ▷ Add bidirectional position encoding
$\mathbf{X} = \text{Dropout}(\mathbf{X})$
Step 2: Bidirectional Depthwise Convolution
$\mathbf{X}_{\text{proj}} = \mathbf{W}_{\text{in}}\mathbf{X} \in \mathbb{R}^{B\times T\times H}$ ▷ Input projection
$\mathbf{X}_{\text{conv}}^{\text{fwd}} = \text{GELU}(\text{Conv1D}_{\text{fwd}}(\mathbf{X}_{\text{proj}}))$ ▷ Forward convolution
$\mathbf{X}_{\text{conv}}^{\text{bwd}} = \text{GELU}(\text{Conv1D}_{\text{bwd}}(\text{Flip}(\mathbf{X}_{\text{proj}})))$ ▷ Backward convolution
$\mathbf{X}_{\text{conv}} = \mathbf{X}_{\text{conv}}^{\text{fwd}} + \mathbf{X}_{\text{conv}}^{\text{bwd}} \in \mathbb{R}^{B\times T\times H}$
Step 3: History Window Construction
$\mathbf{X}_{\text{padded}} = [\mathbf{0}_{B\times L\times H}, \mathbf{X}] \in \mathbb{R}^{B\times(T+L)\times H}$
$\mathbf{H}_{\text{hist}} = \text{Unfold}(\mathbf{X}_{\text{padded}}, \text{window} = L, \text{step} = 1) \in \mathbb{R}^{B\times T\times L\times H}$
Step 4: Soft Semantic Lag Estimation (Algorithm 1)
$(\boldsymbol{\alpha},\ \bar{\mathbf{X}}^{\text{hist}},\ \hat{s}^{\text{soft}}) = \text{LagEstimator}(\mathbf{X},\ \mathbf{H}_{\text{hist}})$
▷ $\boldsymbol{\alpha} \in \mathbb{R}^{B\times T\times L}$, $\bar{\mathbf{X}}^{\text{hist}} \in \mathbb{R}^{B\times T\times H}$, $\hat{s}^{\text{soft}} \in \mathbb{R}^{B\times T}$
Step 5: Compute Input Projection
$\mathbf{b}_k = \mathbf{B}\cdot(\mathbf{X} + \mathbf{X}_{\text{conv}})^\top \in \mathbb{R}^{B\times T\times H}$ ▷ Projected input
Step 6: Compute Delay Correction
$\bar{\mathbf{X}}^{\text{proj}} = \mathbf{C}\cdot\bar{\mathbf{X}}^{\text{hist}} \in \mathbb{R}^{B\times T\times H}$ ▷ Project historical context
$\mathbf{d}_k = \gamma\cdot\tanh\big(\hat{s}^{\text{soft}}\odot\bar{\mathbf{X}}^{\text{proj}}\big) \in \mathbb{R}^{B\times T\times H}$ ▷ Delay correction term
Step 7: Diagonal State Space Recurrence (Zero-Loop)
$\mathbf{a} = \text{diag}(\mathbf{A}) \in \mathbb{R}^{H}$ ▷ Diagonal state transition
$t = [0, 1, \ldots, T-1] \in \mathbb{R}^{T}$ ▷ Time indices
$\mathbf{a}^{\text{pow}}[t,h] = (a_h)^t \in \mathbb{R}^{T\times H}$ ▷ Exponential powers
$\mathbf{b}_{\text{scaled}} = \mathbf{b}_k \odot (\mathbf{a}^{\text{pow}} + \varepsilon)^{-1} \in \mathbb{R}^{B\times T\times H}$
$\mathbf{h}_{\text{scaled}} = \text{CumSum}(\mathbf{b}_{\text{scaled}}, \text{dim} = 1) \in \mathbb{R}^{B\times T\times H}$
$\mathbf{h}_k = \mathbf{h}_{\text{scaled}} \odot \mathbf{a}^{\text{pow}} \in \mathbb{R}^{B\times T\times H}$ ▷ Latent state from recurrence
Step 8: Fuzzy Semantic Modulation (Algorithm 2)
$\mathbf{z} = \dfrac{\mathbf{d}_k}{2} \in \mathbb{R}^{B\times T\times H}$ ▷ Semantic context (approximation)
$(\hat{\mathbf{f}},\ \hat{\mathbf{g}}) = \text{FuzzySystem}(\mathbf{z}) \in \mathbb{R}^{B\times T\times H}$ ▷ Modulation vectors
Step 9: Final State Update
$\mathbf{s}_k = \hat{\mathbf{f}} \odot \text{SiLU}(\mathbf{h}_k) + \hat{\mathbf{g}} \odot \mathbf{d}_k \in \mathbb{R}^{B\times T\times H}$
$\mathbf{s}_k = \text{Dropout}(\mathbf{s}_k)$
Step 10: Residual Connection and Output Projection
$\mathbf{s}_k = \mathbf{W}_{\text{out}}\mathbf{s}_k \in \mathbb{R}^{B\times T\times H}$ ▷ Output projection
$\mathbf{s}_k = \text{LayerNorm}(\mathbf{s}_k + \mathbf{X})$ ▷ Residual connection
Step 11: Extract Output States
$\mathbf{s}_{\text{final}} = \mathbf{s}_k[:, -1, :] \in \mathbb{R}^{B\times H}$ ▷ Final token state
$\bar{\mathbf{s}} = \text{Mean}(\mathbf{s}_k, \text{dim} = 1) \in \mathbb{R}^{B\times H}$ ▷ Average state
return $\mathbf{s}_{\text{final}},\ \bar{\mathbf{s}}$

B.1 Existence and Uniqueness of Solutions

Consider the nonlinear time-delay system

$$\dot{\mathbf{s}}(t) = \phi\big(\mathbf{A}\mathbf{s}(t) + \mathbf{B}\mathbf{x}(t)\big) + \gamma\,\psi\big(\hat{s}(t)\,\mathbf{C}\,\bar{\mathbf{x}}(t-\tau(t))\big), \tag{36}$$

where $\mathbf{s}(t) \in \mathbb{R}^d$ is the state, $\mathbf{x}(t)$ is the input, $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are constant matrices, ϕ, ψ are nonlinear activation functions, $\gamma > 0$ is a gain, $\hat{s}(t) \in [0,1]$ is the similarity score, and $\tau(t)$ is a time-varying delay satisfying $0 \le \tau(t) \le \tau_{\max}$.

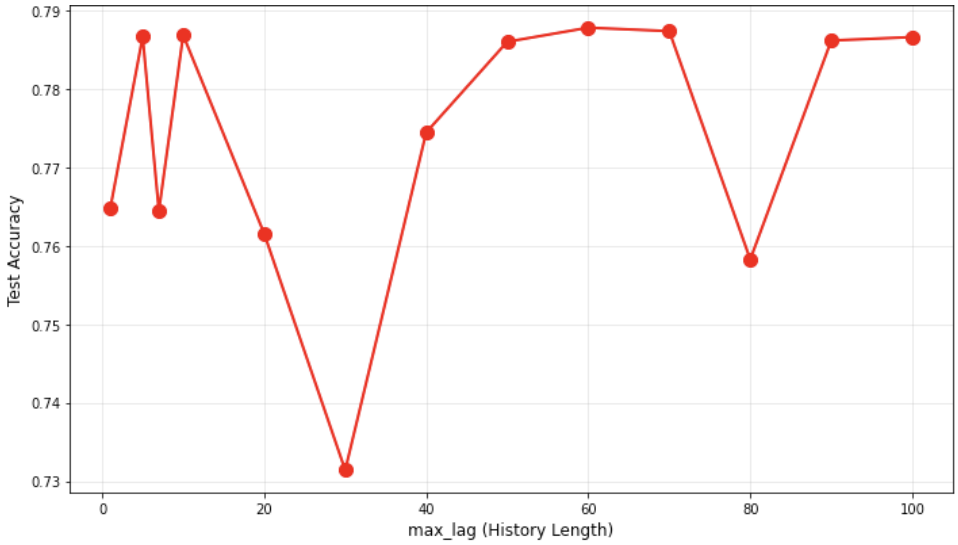

Figure 3: Effects of different lag on IMDB dataset

Theorem 1 (Existence and Uniqueness)**.** *Assume that the nonlinear functions $\phi(\cdot)$ and $\psi(\cdot)$ are Lipschitz continuous with Lipschitz constants L_ϕ and L_ψ, respectively. Then, for any bounded initial history function $\mathbf{s}(t) = \mathbf{s}_0(t)$ defined on $t \in [-\tau_{\max}, 0]$, the nonlinear time-delay system* (36) *admits a unique continuous solution $\mathbf{s}(t)$ on any finite interval $t \in [0, T]$.*

Proof. The proof reformulates the delay differential equation as an integral equation and applies the Banach fixed-point theorem.

Let $\mathcal{C} = C([-\tau_{\max}, T], \mathbb{R}^d)$ be the Banach space of continuous functions equipped with the supremum norm $\|\mathbf{s}\|_\infty = \sup_t \|\mathbf{s}(t)\|$. Define the closed subset

$$\mathcal{B} = \{\mathbf{s} \in \mathcal{C} : \mathbf{s}(t) = \mathbf{s}_0(t) \text{ for } t \in [-\tau_{\max}, 0],\ \|\mathbf{s} - \mathbf{s}_0\|_\infty \le R\}$$

for some $R > 0$ to be determined.

For $t \in [0, T]$, any solution to (36) must satisfy the integral equation

$$\mathbf{s}(t) = \mathbf{s}(0) + \int_0^t \Big[\phi\big(\mathbf{A}\mathbf{s}(u) + \mathbf{B}\mathbf{x}(u)\big) + \gamma\, \psi\big(\hat{s}(u)\, \mathbf{C}\, \bar{\mathbf{x}}(u - \tau(u))\big)\Big] du.$$

Define the operator $\mathcal{F} : \mathcal{B} \to \mathcal{C}$ by

$$(\mathcal{F}\mathbf{s})(t) = \begin{cases} \mathbf{s}_0(t), & t \in [-\tau_{\max}, 0], \\ \mathbf{s}(0) + \displaystyle\int_0^t \Big[\phi\big(\mathbf{A}\mathbf{s}(u) + \mathbf{B}\mathbf{x}(u)\big) + \gamma\, \psi\big(\hat{s}(u)\, \mathbf{C}\, \bar{\mathbf{x}}(u - \tau(u))\big)\Big] du, & t \in [0, T]. \end{cases}$$

For any $\mathbf{s}_1, \mathbf{s}_2 \in \mathcal{B}$ and $t \in [0, T]$, the Lipschitz continuity of ϕ and ψ yields

$$\begin{aligned} \|(\mathcal{F}\mathbf{s}_1)(t) - (\mathcal{F}\mathbf{s}_2)(t)\| \le{}& \int_0^t L_\phi \|\mathbf{A}\| \|\mathbf{s}_1(u) - \mathbf{s}_2(u)\| du \\ &+ \gamma \int_0^t L_\psi \|\mathbf{C}\| \|\bar{\mathbf{x}}\|_\infty |\hat{s}_1(u) - \hat{s}_2(u)| du, \end{aligned}$$

where $\|\bar{\mathbf{x}}\|_\infty = \sup_t \|\bar{\mathbf{x}}(t)\| < \infty$.

Since $\hat{s}(u)$ is a continuous function of $\mathbf{s}(u)$, there exists a constant $L_{\hat{s}} > 0$ such that $|\hat{s}_1(u) - \hat{s}_2(u)| \le L_{\hat{s}} \|\mathbf{s}_1(u) - \mathbf{s}_2(u)\|$. Thus,

$$\|(\mathcal{F}\mathbf{s}_1)(t) - (\mathcal{F}\mathbf{s}_2)(t)\| \le K \int_0^t \|\mathbf{s}_1(u) - \mathbf{s}_2(u)\| du,$$

where $K = L_\phi \|\mathbf{A}\| + \gamma L_\psi \|\mathbf{C}\| \|\bar{\mathbf{x}}\|_\infty L_{\hat{s}}$.

Taking the supremum over $t \in [0, T]$ gives

$$\|\mathcal{F}\mathbf{s}_1 - \mathcal{F}\mathbf{s}_2\|_\infty \leq KT \|\mathbf{s}_1 - \mathbf{s}_2\|_\infty.$$

Choose T sufficiently small so that $KT < 1$. Then $\mathcal{F}$ is a contraction on $\mathcal{B}$. By the Banach fixed-point theorem, there exists a unique $\mathbf{s}^* \in \mathcal{B}$ such that $\mathcal{F}\mathbf{s}^* = \mathbf{s}^*$, which is the unique solution on $[0, T]$.

To extend to the entire interval $[0, T_{\text{final}}]$, repeat the argument. Let $T_1 = T$ and define $\mathbf{s}(t)$ on $[0, T_1]$. Then take the solution at $t = T_1$ as the new initial history and extend to $[T_1, 2T]$. Finitely many such steps yield a unique solution on $[0, T_{\text{final}}]$.

The Lipschitz continuity of ϕ and ψ holds for common activation functions like $\tanh$ and SiLU. The input $\mathbf{x}(t)$ is bounded by construction (e.g., word embeddings), and the initial history $\mathbf{s}_0(t)$ is given and bounded. Hence, all conditions are satisfied. □

B.2 Boundedness of the State Trajectory

Theorem 2 (Boundedness of Solutions)**.** *Assume the following conditions hold:*

1. *The input $\mathbf{x}(t)$ is uniformly bounded: there exists $M_x > 0$ such that $\|\mathbf{x}(t)\| \leq M_x$ for all $t \geq 0$.*
2. *The activation functions ϕ and ψ are Lipschitz continuous with constants $L_\phi, L_\psi > 0$ and satisfy $\phi(\mathbf{0}) = \mathbf{0}$, $\psi(\mathbf{0}) = \mathbf{0}$.*
3. *The similarity score satisfies $0 \leq \hat{s}(t) \leq 1$ for all t.*

Then the state trajectory $\mathbf{s}(t)$ satisfies the following bound for all $t \geq 0$:

$$\|\mathbf{s}(t)\| \leq \|\mathbf{s}(0)\| e^{L_\phi \|\mathbf{A}\| t} + \frac{L_\phi \|\mathbf{B}\| M_x + \gamma L_\psi \|\mathbf{C}\| M_x}{L_\phi \|\mathbf{A}\|} \left(e^{L_\phi \|\mathbf{A}\| t} - 1 \right), \tag{37}$$

provided that $\mathbf{A} \neq \mathbf{0}$. If $\mathbf{A} = \mathbf{0}$, the bound simplifies to

$$\|\mathbf{s}(t)\| \leq \|\mathbf{s}(0)\| + (L_\phi \|\mathbf{B}\| M_x + \gamma L_\psi \|\mathbf{C}\| M_x)\, t. \tag{38}$$

Proof. Take the Euclidean norm on both sides of (8). By the triangle inequality,

$$\|\dot{\mathbf{s}}(t)\| \leq \left\| \phi\big(\mathbf{A}\mathbf{s}(t) + \mathbf{B}\mathbf{x}(t)\big) \right\| + \gamma \left\| \psi\big(\hat{s}(t)\, \mathbf{C}\, \mathbf{x}(t - \tau(t))\big) \right\|.$$

Since ϕ is Lipschitz continuous with $\phi(\mathbf{0}) = \mathbf{0}$, we have

$$\left\| \phi\big(\mathbf{A}\mathbf{s}(t) + \mathbf{B}\mathbf{x}(t)\big) \right\| \leq L_\phi \|\mathbf{A}\mathbf{s}(t) + \mathbf{B}\mathbf{x}(t)\| \leq L_\phi \|\mathbf{A}\| \|\mathbf{s}(t)\| + L_\phi \|\mathbf{B}\| \|\mathbf{x}(t)\|.$$

Similarly, using the Lipschitz continuity of ψ, the boundedness of $\hat{s}(t) \in [0, 1]$, and $\psi(\mathbf{0}) = \mathbf{0}$,

$$\left\| \psi\big(\hat{s}(t)\, \mathbf{C}\, \mathbf{x}(t - \tau(t))\big) \right\| \leq L_\psi \|\mathbf{C}\| \|\mathbf{x}(t - \tau(t))\| \leq L_\psi \|\mathbf{C}\| M_x.$$

Substituting these bounds and using $\|\mathbf{x}(t)\| \leq M_x$, we obtain the differential inequality

$$\|\dot{\mathbf{s}}(t)\| \leq a \|\mathbf{s}(t)\| + b,$$

where $a = L_\phi \|\mathbf{A}\|$ and $b = L_\phi \|\mathbf{B}\| M_x + \gamma L_\psi \|\mathbf{C}\| M_x$.

This inequality can be rewritten as

$$\frac{d}{dt} \left(\|\mathbf{s}(t)\| e^{-at} \right) \leq b e^{-at}.$$

Integrating from 0 to t gives

$$\|\mathbf{s}(t)\| e^{-at} - \|\mathbf{s}(0)\| \leq b \int_0^t e^{-a\tau} d\tau = \frac{b}{a} \left(1 - e^{-at} \right).$$

Multiplying by e^{at} yields

$$\|\mathbf{s}(t)\| \leq \|\mathbf{s}(0)\|e^{at} + \frac{b}{a}\left(e^{at} - 1\right).$$

Substituting back $a = L_\phi\|\mathbf{A}\|$ and $b = L_\phi\|\mathbf{B}\|M_x + \gamma L_\psi\|\mathbf{C}\|M_x$ proves (37). When $\mathbf{A} = \mathbf{0}$, the differential inequality reduces to $\|\dot{\mathbf{s}}(t)\| \leq b$, and direct integration gives (38). □

This result guarantees that under mild assumptions, the system is free from divergence or blow-up, and its trajectories evolve within a bounded region of the latent state space.

B.3 Stability of the Equilibrium

We next examine the stability of the equilibrium point. Let $\mathbf{s}^*$ denote a fixed point satisfying

$$\mathbf{0} = \phi\big(\mathbf{A}\mathbf{s}^* + \mathbf{B}\mathbf{x}^*\big) + \gamma\,\psi\big(\hat{s}^*\,\mathbf{C}\,\bar{\mathbf{x}}^*\big), \tag{39}$$

for some constant input $\mathbf{x}^*$ and delay τ^*.

Theorem 3 (Local Exponential Stability)**.** *If the spectral radius $\rho(\mathbf{A}) < 1$ and the nonlinearities ϕ and ψ satisfy $\|\nabla\phi\| \leq L_\phi$ and $\|\nabla\psi\| \leq L_\psi$, then the equilibrium $\mathbf{s}^*$ is locally exponentially stable; that is, there exist constants $K > 0$ and $\eta > 0$ such that*

$$\|\mathbf{s}(t) - \mathbf{s}^*\| \leq Ke^{-\eta t}\|\mathbf{s}(0) - \mathbf{s}^*\|. \tag{40}$$

Proof. Define the error vector $\mathbf{e}(t) = \mathbf{s}(t) - \mathbf{s}^*$. Assuming $\mathbf{x}(t) = \mathbf{x}^*$ is constant in a neighborhood of the equilibrium, we have $\mathbf{x}(t) = \mathbf{x}^*$ and $\mathbf{x}(t - \tau(t)) = \mathbf{x}^*$. Then $\hat{s}(t) = \hat{s}^* + \Delta\hat{s}(t)$, where $\Delta\hat{s}(t) = \frac{\partial\hat{\mathbf{s}}}{\partial\mathbf{s}}\mathbf{e}(t) + \mathcal{O}(\|\mathbf{e}\|^2)$.

Expanding ϕ and ψ around the equilibrium using Taylor series:

$$\phi\big(\mathbf{A}\mathbf{s}(t) + \mathbf{B}\mathbf{x}^*\big) = \phi\big(\mathbf{A}\mathbf{s}^* + \mathbf{B}\mathbf{x}^*\big) + \nabla\phi(\mathbf{A}\mathbf{s}^* + \mathbf{B}\mathbf{x}^*) \cdot \mathbf{A}\mathbf{e}(t) + \mathcal{O}(\|\mathbf{e}\|^2), \tag{41}$$

$$\psi\big(\hat{s}(t)\,\mathbf{C}\,\mathbf{x}^*\big) = \psi\big(\hat{s}^*\,\mathbf{C}\,\mathbf{x}^*\big) + \nabla\psi(\hat{s}^*\,\mathbf{C}\,\mathbf{x}^*) \cdot \mathbf{C}\,\mathbf{x}^* \cdot \frac{\partial\hat{s}}{\partial\mathbf{s}}\mathbf{e}(t) + \mathcal{O}(\|\mathbf{e}\|^2). \tag{42}$$

$$\mathbf{s}^* = \phi\big(\mathbf{A}\mathbf{s}^* + \mathbf{B}\mathbf{x}^*\big) + \gamma\,\psi\big(\hat{s}^*\,\mathbf{C}\,\bar{\mathbf{x}}^*\big). \tag{43}$$

Substituting these expansions into (8) and using the equilibrium condition (43), we obtain the linearized error dynamics:

$$\dot{\mathbf{e}}(t) = \mathbf{A}_\phi\mathbf{e}(t) + \gamma\mathbf{C}_\psi\mathbf{e}(t - \tau^*), \tag{44}$$

where

$$\mathbf{A}_\phi = \nabla\phi(\mathbf{A}\mathbf{s}^* + \mathbf{B}\mathbf{x}^*) \cdot \mathbf{A}, \qquad \mathbf{C}_\psi = \nabla\psi(\hat{s}^*\,\mathbf{C}\,\mathbf{x}^*) \cdot \mathbf{C}\,\mathbf{x}^* \cdot \frac{\partial\hat{s}}{\partial\mathbf{s}}.$$

From the Lipschitz assumptions, we have the bounds

$$\|\mathbf{A}_\phi\| \leq L_\phi\|\mathbf{A}\|, \qquad \|\mathbf{C}_\psi\| \leq L_\psi\|\mathbf{C}\|\|\mathbf{x}^*\|L_{\hat{s}}.$$

The characteristic equation of the linear delay system (44) is

$$\det\big(\lambda\mathbf{I} - \mathbf{A}_\phi - \gamma e^{-\lambda\tau^*}\mathbf{C}_\psi\big) = 0. \tag{45}$$

For any root λ of (45), taking norms gives

$$|\lambda| \leq \|\mathbf{A}_\phi\| + \gamma\|\mathbf{C}_\psi\| \leq L_\phi\|\mathbf{A}\| + \gamma L_\psi\|\mathbf{C}\|\|\mathbf{x}^*\|L_{\hat{s}}.$$

Since $\rho(\mathbf{A}) < 1$, there exists a matrix norm $\|\cdot\|_*$ such that $\|\mathbf{A}\|_* < 1$. By continuity, for sufficiently small γ or sufficiently small $\|\mathbf{C}_\psi\|$, we also have $\rho(\mathbf{A}_\phi) < 1$. The small-gain condition

$$\|\mathbf{A}_\phi\| + \gamma\|\mathbf{C}_\psi\| < 1 \tag{46}$$

ensures that all characteristic roots satisfy $\Re(\lambda) < -\eta$ for some $\eta > 0$.

To establish exponential stability rigorously, construct the Lyapunov-Krasovskii functional

$$V(\mathbf{e}_t) = \mathbf{e}(t)^\top\mathbf{P}\mathbf{e}(t) + \int_{t-\tau_{\max}}^{t}\int_{\theta}^{t}\|\mathbf{e}(s)\|^2 ds d\theta, \tag{47}$$

where $\mathbf{P} = \mathbf{P}^\top > 0$ is a positive definite matrix. Taking the time derivative along trajectories of (44) and applying Young's inequality yields

$$\dot{V}(\mathbf{e}_t) \leq \mathbf{e}(t)^\top \big(\mathbf{A}_\phi^\top \mathbf{P} + \mathbf{P}\mathbf{A}_\phi + \tau_{\max}\mathbf{I} + \gamma^2 \mathbf{P}\mathbf{C}_\psi \mathbf{C}_\psi^\top \mathbf{P}\big)\mathbf{e}(t) + \gamma^2 \|\mathbf{e}(t-\tau^*)\|^2 - \int_{t-\tau_{\max}}^{t} \|\mathbf{e}(s)\|^2 ds.$$

If there exists $\mathbf{P} \succ 0$ satisfying the Linear Matrix Inequality

$$\mathbf{A}_\phi^\top \mathbf{P} + \mathbf{P}\mathbf{A}_\phi + \tau_{\max}\mathbf{I} + \gamma^2 \mathbf{P}\mathbf{C}_\psi \mathbf{C}_\psi^\top \mathbf{P} + \alpha \mathbf{I} \prec 0 \tag{48}$$

for some $\alpha > 0$, then

$$\dot{V}(\mathbf{e}_t) \leq -\alpha \|\mathbf{e}(t)\|^2.$$

The condition $\rho(\mathbf{A}_\phi) < 1$ guarantees the existence of such a $\mathbf{P}$ by the Lyapunov-Krasovskii theorem for delay systems. Standard results then imply the existence of constants $K \geq 1$ and $\eta > 0$ such that

$$\|\mathbf{e}(t)\| \leq K e^{-\eta t} \|\mathbf{e}(0)\|.$$

Therefore, the exponential stability of the linearized system implies the local exponential stability of the original nonlinear system (8). Thus, there exists a neighborhood $\mathcal{N}$ of $\mathbf{s}^*$ such that for any initial condition $\mathbf{s}(0) \in \mathcal{N}$, the estimate (40) holds. □

B.4 Stability of the Discrete-Time Realization

The discrete model Eq. (12) inherits the stability of its continuous counterpart.

Theorem 4 (Discrete Stability)**.** *If $\rho(\mathbf{A}) < 1$ and the delay correction gain γ satisfies*

$$\gamma < \frac{1 - \rho(\mathbf{A})}{L_\psi \, \|\mathbf{C}\|}, \tag{49}$$

then the discrete nonlinear time-delay system is contractive:

$$\|\mathbf{s}_k - \mathbf{s}^*\| \leq \big(\rho(\mathbf{A}) + \gamma L_\psi \|\mathbf{C}\|\big) \, \|\mathbf{s}_{k-1} - \mathbf{s}^*\|, \tag{50}$$

and thus globally asymptotically stable.

Proof. We analyze the update mapping $\mathcal{F}(\mathbf{s}) = \phi(\mathbf{A}\mathbf{s} + \mathbf{B}\mathbf{x}^*) + \gamma\psi(\hat{s}^* \mathbf{C}\bar{\mathbf{x}}^*)$. At equilibrium, $\mathbf{s}^* = \mathcal{F}(\mathbf{s}^*)$. For any two states $\mathbf{s}$ and $\mathbf{s}'$, the Lipschitz continuity of ϕ gives

$$\|\phi(\mathbf{A}\mathbf{s} + \mathbf{B}\mathbf{x}^*) - \phi(\mathbf{A}\mathbf{s}' + \mathbf{B}\mathbf{x}^*)\| \leq L_\phi \|\mathbf{A}\| \|\mathbf{s} - \mathbf{s}'\|.$$

Since $\rho(\mathbf{A}) < 1$, we can choose a matrix norm $\|\cdot\|_*$ such that $\|\mathbf{A}\|_* < 1$. For common activation functions like $\tanh$ or SiLU, we have $L_\phi = 1$, so the linear part alone would be a contraction. However, the additional term $\gamma\psi(\hat{s}^* \mathbf{C}\bar{\mathbf{x}}^*)$ is constant at equilibrium and does not affect the contraction property. For states away from equilibrium, the similarity score $\hat{s}(\mathbf{s})$ and historical context $\bar{\mathbf{x}}(\mathbf{s})$ vary. Using the Lipschitz properties of ψ, $\hat{s}$, and $\bar{\mathbf{x}}$, we obtain

$$\begin{aligned}\|\mathcal{F}(\mathbf{s}) - \mathcal{F}(\mathbf{s}')\| \leq \|\mathbf{A}\| \|\mathbf{s} - \mathbf{s}'\| \\ + \gamma L_\psi \|\mathbf{C}\| \big(\|\bar{\mathbf{x}}(\mathbf{s}) - \bar{\mathbf{x}}(\mathbf{s}')\| + \|\bar{\mathbf{x}}(\mathbf{s}')\| \, |\hat{s}(\mathbf{s}) - \hat{s}(\mathbf{s}')|\big).\end{aligned}$$

Near the equilibrium, $\|\bar{\mathbf{x}}(\mathbf{s})\| \approx \|\bar{\mathbf{x}}^*\|$ and $\hat{s}(\mathbf{s}) \approx \hat{s}^*$, so the differences satisfy $\|\bar{\mathbf{x}}(\mathbf{s}) - \bar{\mathbf{x}}(\mathbf{s}')\| \approx 0$ and $|\hat{s}(\mathbf{s}) - \hat{s}(\mathbf{s}')| \approx 0$. Hence, locally,

$$\|\mathcal{F}(\mathbf{s}) - \mathcal{F}(\mathbf{s}')\| \leq \big(\|\mathbf{A}\| + \gamma L_\psi \|\mathbf{C}\| \|\bar{\mathbf{x}}^*\| L_{\hat{s}}\big) \|\mathbf{s} - \mathbf{s}'\|,$$

where $L_{\hat{s}}$ is the Lipschitz constant of $\hat{s}$. Since $\|\mathbf{A}\|$ can be made arbitrarily close to $\rho(\mathbf{A})$ by choosing an appropriate norm, the contraction condition becomes

$$\rho(\mathbf{A}) + \gamma L_\psi \|\mathbf{C}\| \|\bar{\mathbf{x}}^*\| L_{\hat{s}} < 1.$$

Typically $L_{\hat{s}} \leq 1$ because $\hat{s}$ is a normalized similarity measure. Thus, the sufficient condition reduces to

$$\gamma < \frac{1 - \rho(\mathbf{A})}{L_\psi \|\mathbf{C}\| \|\bar{\mathbf{x}}^*\|}.$$

When this inequality holds, the mapping $\mathcal{F}$ is a contraction with Lipschitz constant $L = \rho(\mathbf{A}) + \gamma L_\psi \|\mathbf{C}\| \|\bar{\mathbf{x}}^*\| < 1$. By the Banach fixed-point theorem, $\mathcal{F}$ has a unique fixed point $\mathbf{s}^*$, and for any initial $\mathbf{s}_0$, the iterates $\mathbf{s}_{k+1} = \mathcal{F}(\mathbf{s}_k)$ converge exponentially to $\mathbf{s}^*$:

$$\|\mathbf{s}_k - \mathbf{s}^*\| \leq L^k \|\mathbf{s}_0 - \mathbf{s}^*\|.$$

Since the contraction constant L does not depend on the initial condition, the convergence is global. This completes the proof. □

B.5 Discussion

The results above establish the theoretical soundness of the proposed nonlinear time-delay formulation:

- **Existence & Uniqueness** guarantees that the system's solution is well-defined for any bounded history.
- **Boundedness** prevents state divergence under bounded input conditions.
- **Stability** ensures convergence to equilibrium when $\rho(\mathbf{A}) < 1$ and γ is moderate.

These properties provide a solid mathematical foundation connecting continuous nonlinear delay systems with their discrete-time neural realizations used in modern sequence modeling.