Gated State-Space Model for Event Trigger Detection: A Linear-Time Alternative to Transformers

Chaitanya Kirti
Centre for Linguistic Science and Technology
Indian Institute of Technology Guwahati, India
ckirti@iitg.ac.in

Harsh Kumar
Department of Data Science and Applications
Indian Institute of Technology Madras, India
22f3002198@ds.study.iitm.ac.in

Ashish Anand
Department of Computer Science and Engineering
Indian Institute of Technology Guwahati, India
anand.ashish@iitg.ac.in

Prithwijit Guha
Department of Electronics and Electrical Engineering
Indian Institute of Technology Guwahati, India
pguha@iitg.ac.in

***Abstract*—Event trigger detection aims to identify sparse yet decisive tokens that signal event occurrences in text. Most recent approaches rely on Transformer-based architectures with self-attention, which incur quadratic complexity and recompute global interactions at each layer. In this work, we revisit event trigger detection from a state evolution perspective and propose a gated state-space modeling framework. Our approach builds on a selective state-space model backbone and introduces a gated multi-head state refinement mechanism. The gating selectively amplifies event-relevant state updates while suppressing background tokens, enabling persistent contextual modeling without explicit attention. The model is trained end-to-end using a standard token-level classification objective. We evaluate the proposed models on four benchmark datasets: MAVEN, RAMS, ONTOEVENT, and Vrittanta-EN. Experimental results show that both vanilla and gated state-space models achieve performance comparable to strong Transformer-based baselines, including Gemma-2B, SPEECH, and EDM3. Across datasets, the proposed approach matches or slightly exceeds prior baselines in F1 score without relying on attention mechanisms. This work demonstrates that gated state-space modeling provides a principled and scalable alternative to attention-based architectures for event trigger detection, highlighting state evolution as an effective inductive bias for event trigger detection and related event-centric text understanding.**

***Index Terms*—Event trigger detection, state-space models, Mamba, selective state evolution, long-context modeling**

I. INTRODUCTION

Event trigger detection is a primary task in information extraction, and it involves identifying textual expressions that signal the occurrence of events [23]. Accurate trigger detection aids a wide range of applications, such as temporal reasoning, knowledge graph building, and event-based question answering [2]. As such, there has been intensive research in this area, both in document-level and sentence-level scenarios.

State-of-the-art methods for event trigger detection are based on the use of Transformer encoders [3]. Self-attention-based models have proven highly effective through the use of contextualized representations of tokens, which are learned through large-scale pretraining. These models have been successfully used in the prompt-based, supervised, and generative models of event detection, making attention-based models standard choice for this task [4].

Moreover, existing literature has identified several unique characteristics of event trigger detection. Event triggers are typically sparse compared to the document length, often depend on contextual information spread over sentences, and need consistency over narrative structure instead of token-level interactions [5]. These aspects have prompted the exploration of alternative modeling frameworks that focus on contextual accumulation and strategic information integration.

State-space models (SSMs) are one such alternative framework for sequence modeling. Instead of modeling explicit token interactions, SSMs rely on a hidden state that transitions as the input sequence is processed [6]. Recent improvements in selective state-space modeling have demonstrated the effectiveness of these models for long-context sequence modeling tasks with competitive performance and linear-time complexity [7]. These findings collectively indicate that state evolution could provide an additional inductive bias for modeling structured narratives.

Although there is an increasing interest in using SSMs for language modeling, the use of these models for fine-grained information extraction tasks is still limited. Specifically, it is unclear how state-space representations can be extended to model sparse event-centric signals at the token level. To address this, architectural solutions are required that can selectively focus on event-relevant information while retaining the sequential nature of state evolution.

In this paper, we explore the use of state-space models for event trigger detection and introduce a gated multi-head refinement module built on top of a vanilla SSM architecture. Our proposed method models event trigger detection as a state

evolution process, where contextual information is accumulated incrementally and selectively controlled via gating. The multi-head refinement module enables the refinement of the state to be performed simultaneously without incorporating attention mechanisms or altering the dynamics of the state space.

We evaluate both vanilla and gated state-space models on four benchmark datasets i.e. MAVEN, RAMS, ONTOEVENT, and Vrittanta-EN, covering diverse domains and narrative structures. Our experiments show that state-space models achieve performance comparable to strong Transformer-based baselines, including Gemma-2B. Additionally, the gated variant consistently outperforms the Transformer-based baselines.

The contributions of this paper are summarized as follows:

- We present a gated state-space modeling framework for event trigger detection, combining a Mamba-based backbone with multi-head state refinement.
- We provide a systematic evaluation across four benchmark datasets, comparing state-space models with Transformer-based and prompt-based baselines.
- We conduct targeted analyses to characterize when gated refinement helps most, showing larger improvements in long-document and low-trigger-density settings via document-length trends and dataset-level behavior.

II. RELATED WORK

A. Event Trigger Detection

Event trigger detection has long been studied as a fundamental task in information extraction. Early approaches relied on feature engineering and statistical models such as conditional random fields and support vector machines [8]. With the adoption of deep learning, convolutional and recurrent neural networks became popular, enabling improved contextual modeling and generalization [9].

B. Transformer-Based Approaches for Event Detection

The introduction of large pre-trained language models significantly advanced event trigger detection. Models such as BERT and RoBERTa provide strong contextual representations and have been widely adopted across benchmark datasets [10], [11]. These approaches establish strong baselines and remain highly competitive in practice. Transformer architectures dominate current event detection systems due to their ability to model long-range dependencies using self-attention. Numerous studies fine-tune Transformer encoders for token-level trigger classification, achieving state-of-the-art results on datasets such as MAVEN and ONTOEVENT [3].

More recent work has explored alternative formulations within the Transformer paradigm. EDM3 reframes event detection as a multi-task text generation problem using sequence-to-sequence Transformers [13]. Prompt-based approaches utilize large language models by embedding event schemas and task descriptions into the prompts themselves, thus allowing for easy adaptation to different datasets and events [14]. Other studies use graph structures, schema-based attention, or dynamic decoding strategies to enhance trigger identification [15]–[17].

While these methods achieve strong empirical performance, they are primarily based on self-attention and large pre-trained encoders. As a result, recent work has begun to explore complementary modeling paradigms that emphasize alternative sequence representations and inductive biases, particularly for long-context and document-level event detection.

C. State-Space Models for Sequence Modeling

State-space models (SSMs) provide an alternative framework for sequence modeling by maintaining a latent state that evolves over time. Unlike Transformers, SSMs do not compute explicit token-to-token interactions [18]. Recent advances such as S4 and selective state-space models demonstrate that SSMs can achieve competitive performance on long-sequence tasks with linear-time complexity [24].

Selective SSMs, including Mamba, introduce input-dependent gating mechanisms that control how tokens update the latent state. These models have shown strong results in language modeling and time-series analysis while offering favorable efficiency properties [6]. However, their application to fine-grained information extraction tasks, such as event trigger detection, remains limited.

Our approach is based on selective state space modeling, adapted to the problem of event trigger detection. We propose a gated multi-head refinement module that improves the selectivity of the state without changing the dynamics of the state evolution. This enables effective modeling of sparse, context-dependent event triggers while preserving the efficiency advantages of SSMs.

III. PROBLEM DEFINITION

A. Event Trigger Detection Task

Event trigger detection is related to the problem of identifying tokens that serve as explicit indicators of the occurrence of events in text. For a given input document, the goal is to classify each token as belonging to an event trigger or not. In this research, the focus is strictly on trigger detection, irrespective of the event arguments and roles.

Mathematically, a document is modeled as a sequence of tokens. An event trigger is usually realized by a verb or noun that signifies an action, occurrence, or state transition. A document can have zero, one, or more event triggers. Event triggers are usually sparse and may depend on long-range context information over sentences [23].

Event triggers can extend over one or more consecutive tokens. As is conventional in event detection benchmarks like MAVEN [12], RAMS [25], and ONTOEVENT [26], trigger spans extending over multiple tokens are projected to token-level annotations. All tokens in an annotated trigger span are marked as trigger tokens. Therefore, event trigger detection is modeled as a binary token-level classification task. This token-level modeling is consistent with conventional practice in event detection benchmarks.

B. Notation and Formal Setup

Let $X = (x_1, x_2, \ldots, x_L)$ denote an input sequence of L tokens. The objective is to predict a corresponding label sequence $Y = (y_1, y_2, \ldots, y_L)$, where $y_t \in \{0, 1\}$. A label $y_t = 1$ indicates that token x_t is part of an event trigger, while $y_t = 0$ denotes a non-trigger token.

Given an encoder model $f(\cdot)$, the input sequence is mapped to a sequence of contextual representations:

$$H = f(X) = (h_1, h_2, \ldots, h_L), \tag{1}$$

where $h_t \in \mathbb{R}^d$ represents the contextual representation of token x_t.

A classifier $g(\cdot)$ is applied to each hidden state to produce token-level predictions:

$$\hat{y}_t = g(h_t), \tag{2}$$

where $\hat{y}_t$ denotes the predicted probability that token x_t is an event trigger.

The model is trained by minimizing the token-level cross-entropy loss over the labeled dataset. Special tokens and padded positions are excluded from the loss computation. This formulation is consistent with prior Transformer-based and generative approaches to event trigger detection, enabling direct and fair comparison [3], [13].

IV. From Attention to State Evolution

A. Limitations of Attention for Event Detection

Self-attention is the most common form of attention in modern event trigger detection models [19]. By computing the interaction between all pairs of tokens, Transformer-based models are able to capture contextual relationships and achieve strong empirical results. However, this approach also has several drawbacks when used in event-centric modeling.

First, self-attention has a quadratic complexity in the length of the input sequence, which raises efficiency issues in document-level event trigger detection, where triggers can depend on contextually dispersed information across several sentences [20]. In this case, models are often forced to use techniques such as truncation or sliding windows, which can break the narrative flow and coherence of the story.

Second, attention is symmetric across all tokens [21]. However, in event trigger detection, this assumption is suboptimal because event triggers are sparse and semantically significant, whereas attention mechanisms spend equal computational resources on both semantically significant and insignificant tokens. This can cause inefficient use of model capacity, especially when the trigger density is low.

Third, attention mechanisms do not preserve a constant notion of persistent state [22]. Instead, the interaction between tokens is re-computed at each layer, and global context is represented implicitly rather than explicitly. In applications requiring coherence over long narratives, such as event trigger detection, the lack of explicit state preservation can limit the ability of the model to accumulate contextual information.

It is important to note that these points do not argue against the utility of attention-based models. Instead, they imply that other inductive biases might be more suitable for modeling the structural properties of event-centric text.

B. State Evolution as an Alternative Paradigm

State-space models provide an alternative sequence modeling paradigm based on state evolution. Rather than computing explicit token-to-token interactions, they maintain a latent state that is updated incrementally as the input sequence is processed, serving as a compact summary of accumulated context [24]. By defining sequence modeling as the evolution of states over time, these models retain information over long contexts in a linear-time fashion without repeated global recomputation [28]. This inductive bias is very useful for trigger detection in events, where relevant evidence can be limited and fragmented over sentences. Recent advances in selective state-space models show that these models can achieve strong language understanding capabilities with better efficiency [29]. These results provide a strong motivation to use state-space modeling as a scalable choice over attention-based models for event-centric tasks.

V. Gated State-Space Modeling for Event Trigger Detection

A. Vanilla State-Space Model Backbone

We adopt a selective state-space model as the backbone encoder for event trigger detection. Given an input token sequence $X = (x_1, x_2, \ldots, x_L)$, the model processes the sequence in a strictly sequential manner by maintaining a latent state that is updated at each time step.

We use the Mamba-2 selective state-space model [6] as implemented in the `state-spaces/mamba2-2.7b` pretrained checkpoint from Hugging Face. The model maintains a hidden state dimension of $d = 2560$ throughout all layers. Mamba-2 extends the original Mamba architecture with a structured state-space layer (SSD) that improves computational efficiency while preserving selective state evolution. At each position t, the model first computes token-specific parameters through linear projections:

$$\Delta_t = \text{softplus}(\mathbf{W}_\Delta x_t + b_\Delta), \tag{3}$$

$$B_t = \mathbf{W}_B x_t, \tag{4}$$

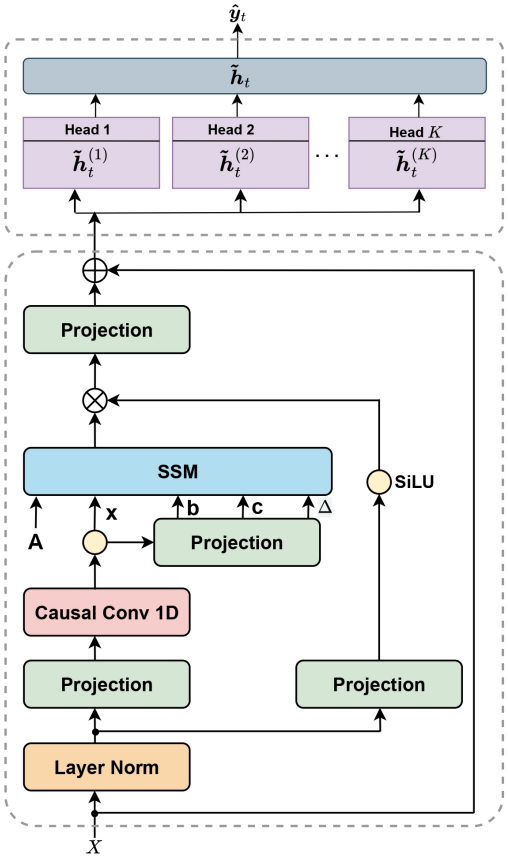
$$C_t = \mathbf{W}_C x_t, \tag{5}$$

where $\mathbf{W}_\Delta \in \mathbb{R}^{d \times d}$, $\mathbf{W}_B \in \mathbb{R}^{d \times N}$, and $\mathbf{W}_C \in \mathbb{R}^{N \times d}$ are learned projection matrices, b_Δ is a learned bias term, and N denotes the state expansion dimension ($N = 64$ in Mamba-2-2.7B). The discretization parameter Δ_t controls the rate at which new input information updates the hidden state, while B_t and C_t govern input-to-state and state-to-output mappings, respectively. The core state evolution follows a discrete-time linear state-space formulation with structured transitions:

$$\bar{h}_t = \mathbf{A}\bar{h}_{t-1} + B_t x_t, \tag{6}$$

$$y_t = C_t \bar{h}_t, \tag{7}$$

Fig. 1. Illustration of the proposed architecture. The lower block depicts the vanilla Mamba state-space backbone. The upper block introduces a gated multi-head refinement module, where multiple parallel gated projections selectively refine the evolving state prior to event trigger prediction.



where $\mathbf{A} \in \mathbb{R}^{N\times N}$ is a structured diagonal transition matrix, $\bar{h}_t \in \mathbb{R}^N$ represents the internal state, and $y_t \in \mathbb{R}^d$ is the output token representation. Unlike Mamba-1, Mamba-2 uses a simplified diagonal structure for $\mathbf{A}$, which enables more efficient parallel computation through matrix chunking and state-space duality [30].

The selective mechanism operates through input-dependent discretization. The final hidden state incorporates the selective gating parameter:

$$h_t = \Delta_t \odot \bar{h}_t, \tag{8}$$

where $\odot$ denotes element-wise multiplication. This selective gating allows the model to modulate how strongly each token's state is retained or updated, enabling adaptive filtering of event-relevant versus background information.

In summary, the parameterized state transition function $\mathcal{F}(\cdot)$ can be formally expressed as:

$$h_t = \mathcal{F}(h_{t-1}, x_t) = \Delta_t \odot \left(\mathbf{A}\bar{h}_{t-1} + B_t x_t\right), \tag{9}$$

where Δ_t, B_t, and C_t are computed according to Equations (3)–(5), and $\mathbf{A}$ is a learned diagonal matrix. We use $\bar{h}_t$ to denote the intermediate state prior to gating, and h_t for the final gated state used for downstream prediction.

B. Gated Multi-Head State Refinement

The backbone state-space model (SSM) provides contextualized token-level states, but it relies on a single shared representation at each layer. To improve the flexibility of the representation, we introduce a gated multi-head state refinement module that is built on top of the outputs of the state-space model. Figure 1 shows the proposed GatedMultiHead-Mamba architecture, focusing on the state-space backbone and the gated multi-head refinement module.

Given the backbone output sequence $H = (h_1, h_2, \ldots, h_L)$, the refinement module applies K parallel gated projections to each state. For the i-th refinement head, the transformed representation is computed as:

$$\tilde{h}_t^{(i)} = \sigma\left(W_g^{(i)} h_t\right) \odot \left(W_p^{(i)} h_t\right), \tag{10}$$

where $W_g^{(i)}$, $W_p^{(i)} \in \mathbb{R}^{d\times d}$ are learnable projection matrices, $\sigma(\cdot)$ denotes the sigmoid activation function, and $\odot$ represents element-wise multiplication.

The gating mechanism modulates the contribution of each projection by controlling how strongly different components of the state are emphasized. The final refined representation is obtained by averaging the outputs of all refinement heads:

$$\tilde{h}_t = \frac{1}{K}\sum_{i=1}^{K} \tilde{h}_t^{(i)}. \tag{11}$$

This refinement strategy increases representational capacity through parallel gated transformations without introducing attention mechanisms or altering the asymptotic computational complexity.

C. Model Architecture Overview

The complete model consists of three stages. First, the input token sequence is encoded using the Mamba-based state-space backbone to produce token-level state representations. Second, the gated multi-head refinement module is applied to each state to obtain refined representations. Finally, a lightweight classification head is used to predict trigger labels at each token position. All components operate in a sequential manner. No attention matrices, key–value caches, or global interaction tensors are constructed. As a result, memory usage remains stable with respect to sequence length.

D. Training Objective

Event trigger detection is formulated as a binary token-level classification task. Each token is labeled as either part of an event trigger or a non-trigger. Given predicted probabilities $\hat{y}_t$ and ground-truth labels y_t, the model is trained using the token-level cross-entropy loss:

$$\mathcal{L} = -\sum_{t=1}^{L} \mathbb{I}(y_t \neq -100)\left[y_t \log \hat{y}_t + (1-y_t)\log(1-\hat{y}_t)\right], \tag{12}$$

where padded or invalid positions are masked using the indicator function $\mathbb{I}(\cdot)$. The model is trained end-to-end using

standard backpropagation. No auxiliary objectives or additional supervision are employed, ensuring a direct and fair comparison with Transformer-based and generative baselines.

VI. Experimental Setup

A. Datasets

We evaluate our models on four established event trigger detection benchmarks: MAVEN, RAMS, OntoEvent, and Vrittanta-EN. These datasets vary in size, event ontology, and contextual scope, providing a diverse evaluation setting.

MAVEN is a large-scale document-level event detection dataset comprising 168 event types. It consists of long news articles with sparse and context-dependent triggers, making it well-suited for evaluating long-range contextual modeling [12].

RAMS focuses on document-level event understanding with rich narrative dependencies. Event triggers in RAMS often rely on information distributed across multiple sentences, posing challenges for local context modeling [25].

OntoEvent contains sentence-level and short document examples derived from ACE-style annotations. It is commonly used to evaluate event detection under limited contextual scope [26].

Vrittanta-EN is a recently introduced English event detection dataset designed to capture diverse narrative styles and domains [27]. It includes both simple and complex event mentions, offering a complementary evaluation scenario.

B. Baselines

We compare the proposed gated state-space model against several strong baselines representing current state-of-the-art approaches. For supervised event trigger detection, we adopt SPEECH [3] as the baseline on the MAVEN and OntoEvent datasets, and EDM3 [13] on the RAMS dataset, as these models represent the best-performing supervised approaches reported for the respective benchmarks.

As an attention-based reference model, we include Gemma-2B, a Transformer-based encoder with 2.5 billion parameters, which is comparable in scale to the Mamba-2-2.7B backbone (2.7 billion parameters). Gemma-2B has demonstrated competitive performance on a range of language understanding tasks and serves as a representative attention-based alternative [31].

To isolate the effect of the proposed gated refinement, we additionally evaluate a vanilla Mamba-based state-space model that uses the same backbone architecture but omits the gated multi-head refinement module. This controlled comparison enables direct assessment of the contribution of gated state refinement independent of the underlying state-space encoder.

All models are trained and evaluated using identical data splits, preprocessing steps, and evaluation protocols to ensure a fair and consistent comparison.

C. Implementation Details

All models are implemented in PyTorch using the HuggingFace Transformers framework. For the state-space backbone, we use the pretrained `state-spaces/mamba2-2.7b` checkpoint and fine-tune it separately on each dataset. The Mamba-2 architecture provides comparable model capacity to Gemma-2B (2.7B vs. 2.5B parameters), ensuring a fair comparison between state-space and attention-based approaches.

The gated multi-head refinement module is applied on top of the final hidden states produced by the state-space encoder. We set the number of gated refinement heads to $K = 4$. Increasing the number of heads improves representational diversity but introduces linear computational overhead. Preliminary experiments with $K \in \{1, 2, 4, 8\}$ showed that F1 gains over $K = 2$ were marginal (< 0.3 points) for $K \geq 4$, so we adopt $K = 4$ as an efficient trade-off across all datasets. Models are optimized using the AdamW optimizer with a learning rate of 2e-5. Training is performed for 10 epochs, with early stopping based on the development set F1 score. Batch sizes of 4 are selected based on GPU memory constraints. All experiments are conducted on NVIDIA A100 GPUs. We report results on the standard test splits provided by each dataset.

VII. Results and Analysis

A. Main Results Across Datasets

Table I reports token-level Precision, Recall, and F1 scores across four event trigger detection benchmarks. We compare the vanilla Mamba backbone, the proposed Gated Multi-Head Mamba, and representative Transformer-based baselines. Paired bootstrap testing with 1,000 samples confirms that the improvements of the gated state-space model over the vanilla Mamba backbone are statistically significant across all datasets.

Across all datasets, state-space models achieve strong and consistent performance. On MAVEN, the gated model attains an F1 score of 0.7947, improving over the vanilla variant and closely matching Transformer-based baselines. On OntoEvent, the gated model achieves an F1 of 0.7752, exceeding both the vanilla state-space model and previously reported SPEECH-based results.

The largest improvement is observed on RAMS. Here, the gated model increases F1 from 0.6749 to 0.6964, matching or slightly surpassing strong generative baselines such as EDM3. On Vrittanta-EN, Mamba performed better, with the gated variant achieving the highest F1 score of 0.9089. These results indicate that state-space models generalize well across datasets with varying document length, trigger density, and narrative structure.

B. Comparison with Transformer and Prompt-based Baselines

We compare our approach against Transformer-based and prompt-based baselines, including SPEECH, EDM3, EventPrompt, and Gemma-2B. Across all datasets, the gated state-space model achieves performance that is comparable to or better than these baselines. On MAVEN, the gated model closely matches SPEECH and Gemma-2B in F1 score. On RAMS, it slightly outperforms EDM3. On OntoEvent, the gated model achieves the highest reported F1 among all evaluated methods. Performance on Vrittanta-EN performed better

TABLE I
EVENT TRIGGER DETECTION RESULTS ACROSS DATASETS. P, R, AND F1 DENOTE PRECISION, RECALL, AND F1 SCORE RESPECTIVELY.

Model	MAVEN			RAMS			ONTOEVENT			Vrittanta-EN		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Baseline	0.7882	0.7937	0.7909	–	–	0.6953	0.7467	0.7473	0.7470	0.8247	0.8830	0.8524
Gemma-2B	0.7783	0.8016	0.7898	0.7405	0.6565	0.6960	0.7713	0.7590	0.7651	0.8249	0.9017	0.8616
Vanilla-Mamba	0.7683	0.8163	0.7915	0.7545	0.6104	0.6749	0.7617	0.7746	0.8840	0.8319	0.9110	0.8976
GatedMultiHead-Mamba	0.7754	0.8150	**0.7947**	0.7157	0.6781	**0.6964**	0.8197	0.7353	**0.7752**	0.8980	0.9240	**0.9089**

Statistical significance was evaluated using paired bootstrap resampling with 1,000 samples on token-level predictions. For the GatedMultiHead-Mamba model, the 95% confidence intervals for F1 are: MAVEN $\pm$ 0.0069, RAMS $\pm$ 0.0081, ONTOEVENT *$\pm$ 0.0074, and Vrittanta $\pm$ 0.0056. Improvements over the vanilla Mamba baseline are statistically significant ($p < 0.01$ for MAVEN and* ONTOEVENT*, $p < 0.001$ for RAMS, and $p < 0.05$ for Vrittanta).*

than prompt-based and transformer approaches. These results demonstrate that attention-free state-space models can match the accuracy of Transformer and prompt-based systems despite relying on a fundamentally different modeling paradigm.

C. Vanilla vs. Gated State-Space Models

We next compare the vanilla Mamba backbone with the proposed GatedMultiHead-Mamba to assess the effect of gated state refinement. As shown in Table I, the gated variant consistently matches or improves F1 performance across all datasets.

The magnitude of improvement varies by dataset. On RAMS, the gated model yields a substantial F1 gain. On MAVEN and ONTOEVENT, the improvement is more modest but consistent. On Vrittanta-EN, both variants perform similarly, with the gated model achieving the highest overall F1. These results indicate that gated refinement provides a robust performance benefit without degrading accuracy on any dataset.

D. Mechanistic Case Studies

To understand why GatedMultiHead-Mamba behaves differently from Gemma-2B, we analyze representative sentences from each dataset and relate model behavior to the gating mechanism. The examples below use dataset sentences, with trigger words highlighted.

Case 1: Long-range persistence in attribution-heavy narratives (RAMS). This RAMS passage requires maintaining context across multiple sentences with delayed causal resolution: *"In addition to Russian accusations, Syrian Information Minister Omran Zoabi also recently **alleged** that Turkey downed the Russian bomber over Syria . . . "* followed by *"Mr. al-Zoubi **said** in an interview . . . "* and concluding with *"Russia began **delivering** airstrikes . . . and **destroyed** more than 500 trucks."* GatedMultiHead-Mamba correctly identifies triggers such as *alleged*, *said*, and *destroyed* by preserving attribution state across quotations and explanatory clauses. Gemma-2B more frequently misses later triggers when supporting evidence is separated by long narrative spans.

Case 2: Stable trigger tracking in dense event chains (MAVEN). MAVEN narratives often contain tightly coupled sequences of actions, as in the Stockholm siege: *"The Swedish forces had for a long time **laid siege** to Stockholm . . . "*, followed by *"the negotiations were **resumed**"*, and *"the **capitulation** . . . was officially **signed**."* Here, GatedMultiHead-Mamba consistently captures triggers such as *laid siege*, *resumed*, and *signed* without oscillation across adjacent sentences. The gating mechanism helps maintain continuity in the evolving event trajectory, reducing inconsistent trigger detection in dense narratives.

Case 3: Lexical variation under shared escalation structure (ONTOEVENT). ONTOEVENT includes structurally similar escalation patterns with different surface forms, e.g., *"Finland **refused**, so the USSR **invaded** the country."* and *"Israel **refused** and **launched** a large-scale . . . campaign."* Despite lexical differences, GatedMultiHead-Mamba correctly identifies escalation triggers (*invaded*, *launched*) while retaining the causal role of *refused*. This suggests that gating regulates which aspects of prior state are preserved versus reset, supporting generalization across related event structures.

As a general observation, gating supports stable tracking across long attribution chains (RAMS), dense multi-sentence action sequences (MAVEN), and lexically diverse escalation patterns (ONTOEVENT). On Vrittanta-EN, Gemma-2B and GatedMultiHead-Mamba achieve comparable performance, indicating that when triggers are frequent and locally supported, both attention-based and state-space models are equally effective. To validate these qualitative observations, we examined 100 examples where the two models produced different predictions (50 from RAMS and 50 from MAVEN). We categorized these cases by error type. For long-range dependencies—cases where the trigger was separated from critical context by more than 20 tokens—GatedMamba correctly identified 18 out of 25 triggers, while Gemma-2B identified only 7. However, the pattern reversed for negation contexts: Gemma-2B correctly handled 19 out of 25 such cases, whereas GatedMamba succeeded in only 11.

E. Gating Design Choices

To validate the proposed gating mechanism, we compare it against alternative design choices on the MAVEN and RAMS datasets. Table II reports results for four gating variants:

The proposed sigmoid-based gating consistently outperforms alternatives. Highway gating provides modest improvements but underperforms the proposed design, likely because

TABLE II
ABLATION STUDY ON GATING MECHANISM DESIGNS. ALL VARIANTS USE THE SAME MAMBA-2 BACKBONE WITH $K = 4$ HEADS.

Gating Design	MAVEN F1	RAMS F1
No gating (direct projection)	0.7915	0.6749
Proposed: $\sigma(W_g^{(i)} h_t) \odot (W_p^{(i)} h_t)$	**0.7947**	**0.6964**
Highway: $g \odot W_p h + (1 - g) \odot h$	0.7923	0.6891
Multiplicative: $W_g h \odot W_p h$	0.7901	0.6812

TABLE III
F1 SCORE ACROSS DOCUMENT LENGTH BINS ON THE RAMS DATASET.

Document Length (tokens)	GatedMultiHead-Mamba	Gemma-2B
0–128	0.6950	0.6920
128–256	0.6932	0.6845
256–512	0.6891	0.6623
512–1024	0.6834	0.5987
1024+	0.6718	0.5412

the residual connection dilutes event-specific state updates. Multiplicative gating without sigmoid activation causes unstable gate values during training, leading to poor performance. These results confirm that the proposed gating mechanism achieves a good trade-off between selectivity and stability.

F. Performance Across Document Lengths

To gain an empirical understanding of the effect of sequence length on the performance of the models, we plot the F1 scores as a function of document length on the RAMS dataset. The documents are grouped into bins based on their token length, and the results are shown separately for each bin.

As can be seen in Table III, both models perform similarly on short documents. However, as the length of the document increases, the Transformer-based baseline shows a sharper drop in F1 scores. On the other hand, the gated state-space model shows a more gradual drop-off in performance, even for documents longer than 512 tokens. This observation is consistent with the linear-time evolution of states in state-space models, which does not suffer from quadratic interactions among tokens and is thus better suited for modeling longer contexts.

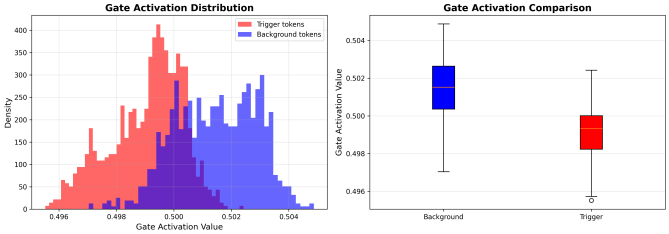
VIII. ABLATION AND ANALYSIS

While Section VII presented empirical performance comparisons, this section provides a deeper analysis of the proposed gating mechanism, its design choices, and its internal behavior.

A. Gating Design Choices

To validate the effectiveness of the proposed gated multi-head refinement module, we compare it against several alternative design variants on the MAVEN and RAMS datasets. All variants use the same Mamba-2 backbone with $K = 4$ heads to ensure a controlled comparison. Table II summarizes the results. The proposed sigmoid-based gating formulation consistently achieves the best performance across both datasets. In contrast, the highway-style gating introduces a residual pathway that weakens the selectivity of event-specific updates, resulting in smaller improvements. The purely multiplicative variant, which omits the sigmoid activation, leads to unstable scaling of state updates during training and degrades overall performance. These findings indicate that the sigmoid gating mechanism provides an effective balance between selectivity and stability. It enables the model to modulate state updates in a controlled manner without introducing excessive variance or suppressing useful contextual information.

Fig. 2. Distribution of gate activation values for trigger and background tokens, illustrating selective and non-saturating state modulation.



B. Performance Across Document Lengths

To examine how sequence length affects model performance, we analyze F1 scores across document length bins on the RAMS dataset. Table III presents the results. Both models perform similarly on short documents. However, as document length increases, the Transformer-based baseline shows a sharper decline in performance. In contrast, the gated state-space model exhibits a more gradual degradation, maintaining relatively stable performance even for long sequences. This trend is consistent with the underlying design of state-space models, which process sequences in linear time without explicit token-to-token interactions. As a result, the model is less sensitive to increases in sequence length and can better preserve contextual information over extended narratives.

C. Gate Activation Analysis

To further understand the behavior of the gating mechanism, we analyze the distribution of gate activation values for trigger and background tokens. Figure 2 illustrates the activation patterns. The results show a consistent separation between trigger and background tokens. Trigger tokens tend to exhibit more concentrated activation values, while background tokens display higher variance. This suggests that the gating mechanism selectively modulates state updates for event-relevant tokens rather than uniformly scaling all inputs. Additionally, gate values are centered around intermediate ranges and do not saturate. This indicates that the gating mechanism operates as a soft modulation function, allowing continuous adjustment of state contributions rather than hard filtering.

Qualitative inspection further reveals that the gating process emphasizes contextually relevant signals while maintaining stable state evolution dynamics. When errors occur, they are typically associated with semantically ambiguous tokens, suggesting that the limitations arise from data ambiguity

rather than instability in the gating mechanism. Overall, these analyses confirm that the proposed gated refinement module improves selective state modulation in a controlled and stable manner, complementing the underlying state-space architecture without increasing computational complexity.

IX. Conclusion

We propose a gated state-space modeling paradigm for the task of event trigger detection, which replaces attention-driven interaction with state evolution. Our method is based on a standard state-space architecture, fortified with gated multi-head state refinement to focus on relevant information for events. Experimental results on four benchmark datasets show that our model achieves comparable performance to strong Transformer-based baselines. The gated version of our model is shown to improve performance consistently, especially for datasets with long documents and few event triggers. In addition to its competitive performance, our model also has desirable efficiency characteristics. The linear-time processing of sequences makes state-space models preferable for document-level event trigger detection. Our findings collectively show that state evolution is a useful inductive bias for event modeling and a scalable, attention-free solution for event extraction. Although the current study focuses on token-level trigger detection, incorporating gated state-space modeling to cover the entire task of event extraction, including argument detection, is a promising direction for future research.

Acknowledgements

The authors used Grammarly for grammar checking and minor language refinement to improve the clarity and readability of this manuscript.

References

[1] Y. Chen, Z. Ding, Q. Zheng, Y. Qin, R. Huang, and N. Shah, "A history and theory of textual event detection and recognition," *IEEE Access*, vol. 8, pp. 201371–201392, 2020.

[2] Y. Zeng, X. Hou, X. Wang, and J. Li, "Advancing logic-driven and complex event perception frameworks for entity alignment in knowledge graphs," *Electronics*, vol. 14, no. 4, Art. no. 670, 2025.

[3] S. Deng, S. Mao, N. Zhang, and B. Hooi, "SPEECH: Structured prediction with energy-based event-centric hyperspheres," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2023, pp. 351–363.

[4] T. Kuculo, S. Abdollahi, and S. Gottschalk, "Transformer-based architectures versus large language models in semantic event extraction: Evaluating strengths and limitations," *Semantic Web*, vol. 16, no. 5, Art. no. 22104968251363759, 2025.

[5] S. Shaar, W. Chen, M. Chatterjee, B. Wang, W. Zhao, and C. Cardie, "Are triggers needed for document-level event extraction?," *Trans. Assoc. Comput. Linguistics*, vol. 13, pp. 1560–1577, 2025.

[6] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," in *Proc. First Conf. on Language Modeling*.

[7] S. Mitra, R. Karami, H. Xu, S. Huang, and H. Kwon, "Characterizing state space model (SSM) and SSM-transformer hybrid language model performance with long context length," *arXiv preprint arXiv:2507.12442*, 2025.

[8] W. Xiang and B. Wang, "A survey of event extraction from text," *IEEE Access*, vol. 7, pp. 173111–173137, 2019.

[9] T. H. Nguyen, K. Cho, and R. Grishman, "Joint event extraction via recurrent neural networks," in *Proc. 2016 Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol. (NAACL-HLT)*, 2016, pp. 300–309.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. 2019 Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol. (NAACL-HLT)*, 2019, pp. 4171–4186.

[11] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[12] X. Wang *et al.*, "MAVEN: A massive general domain event detection dataset," in *Proc. EMNLP*, 2020, pp. 1652–1671.

[13] U. Anantheswaran, H. Gupta, M. Parmar, K. Pal, and C. Baral, "EDM3: Event detection as multi-task text generation," in *Proc. 13th Joint Conf. Lexical and Comput. Semantics (*SEM)*, 2024, pp. 438–451.

[14] Z. Yue, H. Zeng, M. Lan, H. Ji, and D. Wang, "Zero- and few-shot event detection via prompt-based meta learning," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2023, pp. 7928–7943.

[15] L. Li, L. Jin, Z. Zhang, Q. Liu, X. Sun, and H. Wang, "Graph convolution over multiple latent context-aware graph structures for event detection," *IEEE Access*, vol. 8, pp. 171435–171446, 2020.

[16] Z. Xu, P. Wang, W. Ke, G. Li, J. Liu, K. Ji, X. Chen, and C. Wu, "Incorporating schema-aware description into document-level event extraction," in *Proc. Thirty-Third Int. Joint Conf. Artif. Intell. (IJCAI)*, 2024, pp. 6597–6605.

[17] H. Chen, "Dynamic grid tagging scheme for event causality identification and classification," *IEEE Trans. Audio, Speech, Lang. Process.*, 2025.

[18] A. Gu, K. Goel, and C. Re, "Efficiently modeling long sequences with structured state spaces," in *Proc. Int. Conf. Learn. Representations (ICLR)*.

[19] A. Upadhyay, Y. K. Meena, and G. S. Chauhan, "SatCoBiLSTM: Self-attention based hybrid deep learning framework for crisis event detection in social media," *Expert Syst. Appl.*, vol. 249, Art. no. 123604, 2024.

[20] M.-H. Guo, Z.-N. Liu, T.-J. Mu, and S.-M. Hu, "Beyond self-attention: External attention using two linear layers for visual tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 5436–5447, 2022.

[21] Y. Tian, Y. Wang, B. Chen, and S. S. Du, "Scan and snap: Understanding training dynamics and token composition in 1-layer transformer," *Adv. Neural Inf. Process. Syst.*, vol. 36, pp. 71911–71947, 2023.

[22] D. Soydaner, "Attention mechanism in neural networks: Where it comes and where it goes," *Neural Comput. Appl.*, vol. 34, no. 16, pp. 13371–13385, 2022.

[23] Y. Chen, Z. Ding, Q. Zheng, Y. Qin, R. Huang, and N. Shah, "A history and theory of textual event detection and recognition," *IEEE Access*, vol. 8, pp. 201371–201392, 2020.

[24] X. Wang, S. Wang, Y. Ding, Y. Li, W. Wu, Y. Rong, W. Kong, J. Huang, S. Li, H. Yang, *et al.*, "State space model for new-generation network alternative to transformers: A survey," *CoRR*, 2024.

[25] S. Ebner, P. Xia, R. Culkin, K. Rawlins, and B. Van Durme, "Multi-sentence argument linking," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2020, pp. 8057–8077.

[26] S. Deng, N. Zhang, L. Li, C. Hui, H. Tou, M. Chen, F. Huang, and H. Chen, "OntoED: Low-resource event detection with ontology embedding," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2021, pp. 2828–2839.

[27] C. Kirti, A. Chattopadhyay, A. Anand, and P. Guha, "Enhancing event extraction from short stories through contextualized prompts," *arXiv preprint arXiv:2412.10745*, 2024.

[28] J. T. H. Smith, A. Warrington, and S. W. Linderman, "Simplified state space layers for sequence modeling," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2023.

[29] N. Muca Cirone, A. Orvieto, B. Walker, C. Salvi, and T. Lyons, "Theoretical foundations of deep selective state-space models," *Adv. Neural Inf. Process. Syst.*, vol. 37, pp. 127226–127272, 2024.

[30] T. Dao and A. Gu, "Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2024, pp. 10041–10071.

[31] Gemma Team, M. Riviere, S. Pathak, P. G. Sessa, C. Hardin, S. Bhupatiraju, L. Hussenot, T. Mesnard, B. Shahriari, A. Rame, *et al.*, "Gemma 2: Improving open language models at a practical size," *arXiv preprint arXiv:2408.00118*, 2024.