



BALEEN: SELF-INTERPRETABLE, ROBUST SSMS WITH STOCHASTIC SELECTIVE MEMORY

Anonymous authors

Paper under double-blind review

ABSTRACT

We introduce *Baleen*, a family of state space models that unifies **stochastic selection** with **information bottleneck** to build interpretable and robust long-context learners. Unlike Mamba/Mamba2’s deterministic gates, Baleen treats selection as a random variable and regularizes it with a closed-form KL to a sparsity prior: (i) *Baleen-B* samples Bernoulli state-transition gates; (ii) *Baleen-E* samples Exponential time-intervals. This yields an explicit trade-off between retention and compression and exposes token-level selection heatmaps at inference for self-interpretation. On language benchmarks, Baleen improves average accuracy over Mamba2 by +0.95 at 370M pretraining and +1.38 at 7B finetuning. Baleen delivers stronger robustness to localized perturbations and adversarial attacks: under CIFAR-10 sequence perturbation, prefix damage falls to 0.6% vs 26.5% for Mamba2 (average under attacks 0.542 vs 0.385). Finally, Baleen’s self-interpretations outperform IG/Grad-CAM on average fidelity across four text classification tasks. We will release our Baleen-7B models on Hugging Face with code, checkpoints, and an interactive selection-heatmap demo.

1 INTRODUCTION

As competitive successors to transformers (Vaswani et al., 2017), State Space Models (SSMs) have emerged as a powerful neural architecture for sequence-to-sequence modeling, demonstrating impressive efficiency in processing long sequential data (Gu et al., 2021a; Gu & Dao, 2023; Dao & Gu, 2024). SSMs are derived from discretizing a linear time-invariant dynamical system (Gu et al., 2020; 2021b; 2022b). Each step t corresponds to a token and involves two key components: *i*) a fixed-dimension hidden *memory state* matrix $\mathbf{H}_{t-1}$, which maintains a running summarization of past token embeddings $\mathbf{X}_{1:t-1}$ for prediction, and *ii*) a parameterized recurrence, which applies a *state transition* matrix $\mathbf{A}_t$ to transform memory states while sequentially integrating per-token representations into the memory. Recent innovations in SSMs sought to refine the recurrence mechanism to effectively compress context of varying length into a fixed-size memory (Yang et al., 2023; 2024b; Sun et al., 2023; Liu et al., 2024; Sun et al., 2024).

Linear time-variant SSMs (Gu et al., 2021a) have exhibited less satisfactory performance in natural language modeling. The seminal work *Mamba* (Gu & Dao, 2023) introduces a selection mechanism that conditions recurrence parameters on the input sequence with non-linearity. This selection mechanism is designed to enable more flexible context filtering and dynamic selection of important tokens by leveraging the state transition matrix as an input-adaptive gating function, where the parametrization of $\mathbf{A}_t$ involves the input token embedding $\mathbf{X}_t$.

However, since Mamba is trained solely to minimize the prediction error at the population level, there is no explicit constraint enforcing it to retain only the useful context in the memory state. As a result, the memory may capture spuriously correlated information from the context, reducing its capacity to accommodate new, relevant information during recurrence, and deteriorating vulnerability to noisy perturbations (Wang et al., 2024; Poli et al., 2024; Park et al., 2024). Meanwhile, although Mamba is intended to be selective, extracting interpretable patterns to indicate what tokens matter remains challenging. Recent works (Ali et al., 2024; Jafari et al., 2024) attempt to explain Mamba by recovering token-wise importance weights. However, it seems these methods struggle to provide reliable interpretation in language tasks, as Mamba overwhelmingly prioritizes local tokens (Wang et al., 2024).

To address these limitations, we introduce *Baleen*, a family of information-bottleneck SSM architectures that explicitly optimizes for the context compression rate through an stochastic selection mechanism, thereby enabling better generalization, robustness to noisy perturbations, and intrinsic interpretability via stochastic selection gates. Grounded in the information bottleneck principle (Tishby et al., 2000), we conceptualize the hidden memory states in SSMs as an information bottleneck, requiring them to achieve a minimal representation of past context while retaining sufficient information for accurate predictions. From this formulation, we derive two novel SSM architectures equipped with stochastic selection mechanisms and an associated training objective, instantiated through two alternative strategies for randomness modeling.

- **Principled framework of IB-gated SSM with stochastic selection:** We introduce *Baleen* model family, based on a novel state space information bottleneck formulation that converts Mamba’s deterministic selection into stochastic selection with a tractable variational objective. We implement two efficient, plug-in variants: Bernoulli gating of state-transition entries and Exponential timestep sampling, each with a closed-form KL regularizer, preserving the linear-time Mamba kernel.
- **Self-interpretation by design, not post-hoc:** Treating selection as a random variable yields token-level selection heatmaps from expected gates, requiring no extra forward/backward passes and enabling faithful top- k rationales. We also provide an information-theoretic critique of Mamba: its MLE objective maximizes $\mathbb{I}(Y; H)$ without penalizing $\mathbb{I}(H; X)$, while Baleen explicitly trades off these terms.
- **Strong prediction accuracy, higher fidelity of interpretation, and robustness to adversarial attack:** across 11 language tasks, Baleen outperforms Mamba2 at both 370M (+0.95 avg) and 7B (+1.38 avg) and surpass all other baselines. Baleen’s self-interpretations surpass IG/Grad-CAM/Grad×Input in average fidelity on SNLI/IMDb/SST2/RT. Under pixel-sequence perturbations and adversarial prompt attacks ranging from the character level to the sentence level, Baleen demonstrates markedly greater robustness (e.g., prefix [0:32] drop 0.57% vs 26.51% for Mamba2; attack-averaged accuracy 0.542 vs 0.385 on CIFAR-10).

2 PRELIMINARIES

Mamba. In this work, we focus on discrete-time SSMs with real-valued diagonal state transition matrices and zero-order hold discretization rule, a design that underpins the recent success of SSM-based LLMs (Gupta et al., 2022; Gu et al., 2022a; Gu & Dao, 2023).

Let $X := [x_1^\top, \dots, x_T^\top]^\top \in \mathbb{R}^{T \times D}$ be an input sequence, where T is the sequence length and D is the token embedding dimension. A general SSM layer introduces a group of parameters $\{(A_t, B_t, C_t, \Delta_t)\}_{t \in [T]}$ to process the sequence according to the following equations:

$$H_t = A_t \odot H_{t-1} + B_t \odot (1_N x_t^\top), \quad \hat{y}_t = C_t^\top H_t \quad (1)$$

where $\odot$ denotes element-wise multiplication, $A_t = \exp(\hat{A}_t \text{diag}(\Delta_t))$, $B_t = \hat{B}_t \Delta_t^\top$, $\Delta_t \in \mathbb{R}_+^D$, $\hat{A}_t \in \mathbb{R}_{-}^{N \times D}$, $\hat{B}_t \in \mathbb{R}^N$, $C_t \in \mathbb{R}^{N \times D}$ for all $t \in [T]$. Note that Δ_t is strictly positive plus $\hat{A}_t$ is strictly negative, ensuring $A_t \in (0, 1)^{N \times D}$. Define $H := [H_t \in \mathbb{R}^{N \times D}]_{t \in [T]}$ as a $T \times N \times D$ tensor to represent the intermediate *memory states*, and $\hat{Y} := [\hat{y}_1^\top, \dots, \hat{y}_T^\top]^\top \in \mathbb{R}^{T \times D}$ denotes the output sequence. Next, the operation $B_t \odot (1_N x_t^\top)$ encodes the input tokens into the hidden state space. And finally, $C_t^\top H_t$ decodes the memory state to generate the prediction $\hat{y}_t$ for the t -th token.

In *S4* (Gu et al., 2021a), the parameters $\{(\hat{A}_t, \hat{B}_t, C_t, \Delta_t)\}_{t \in [T]}$ are directly learned and remain constant across different token positions. While this approach is effective for certain long-sequence tasks (Gu et al., 2020; 2022b), the time-invariant linearity limits its ability to capture more complex and dynamic signals within the context. Subsequently, *Mamba* (Gu & Dao, 2023) conditions the parameters on the input sequence itself. To be specific, Mamba takes the following form of parameterization:

$$\hat{A}_t = A, \quad \hat{B}_t = W_B x_t, \quad C_t = (W_C x_t) 1_D^\top, \quad \Delta_t = \text{softplus}(W_\Delta x_t), \quad (2a)$$

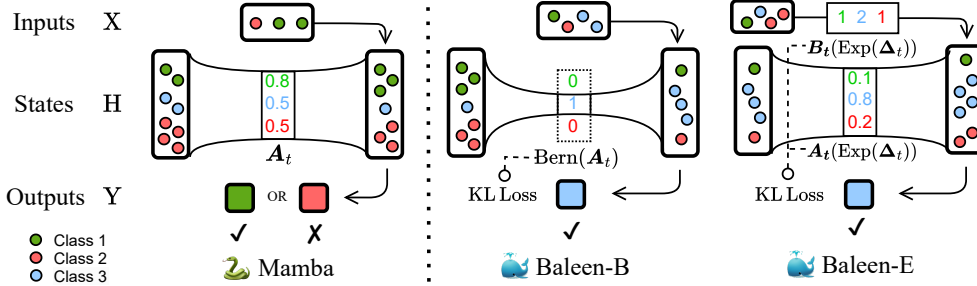


Figure 1: Overview of Baleen compared with previous SSMs such as Mamba. The information bottleneck injects random noise and enhances the selection of the state transition matrix by adapting the deterministic A_t or Δ_t to the randomly sampled variable. For simplicity, we omit deterministic B_t and C_t .

where the learnable parameters in an SSM layer includes $A \in \mathbb{R}^{N \times D}$ being invariant across tokens, $W_B \in \mathbb{R}^{N \times D}$, $W_C \in \mathbb{R}^{N \times D}$, and $W_\Delta \in \mathbb{R}^{D \times D}$. In Mamba2, Dao & Gu (2024) proposes to impose additional structures on the parameter space. Specifically, A is simplified to $\mathbf{1}_N \alpha^\top$ for some $\alpha \in \mathbb{R}^D$ and W_Δ becomes a row-block matrix.

Information Bottleneck. The information bottleneck principle (Tishby et al., 2000) offers a principled approach to balancing the trade-off between representation compactness and predictive capability. It relies on the concept of minimal sufficient statistics to encode information about a target variable $Y \in \mathcal{Y}$ into a representation $Z = \Phi(X) \in \mathcal{Z}$ derived from input $X \in \mathcal{X}$. The principle enforces a regularization on Z by minimizing the mutual information $I(X; Z)$ between the input and the representation, while maximizing the mutual information $I(Y; Z)$ between the representation and the target. Formally, the objective is given by:

$$\min_{\Theta, \Phi} -\mathbb{I}(Y; Z) + \beta \mathbb{I}(Z; X) \quad (3)$$

where β is a hyperparameter to control the balance, Z follows conditional distribution $\mathbb{P}^\Phi(Z|X)$ parametrized by an encoder Φ and Y follows conditional distribution $\mathbb{P}^\Theta(Y|H)$ parametrized by a decoder Θ .

Directly estimating the loss function in Eq. 3 can be computationally infeasible. Previous work Tishby et al. (2000); Alemi et al. (2016); Wu et al. (2020) have derived a tractable variation upper bound (up to a constant) as below:

$$\min_{\Theta, \Phi} -\mathbb{E} [\log \mathbb{P}^\Theta(Y|Z)] + \beta \mathbb{E} [D_{\text{KL}}(\mathbb{P}^\Phi(Z|X) \| \mathbb{Q}(Z))] \quad (4)$$

3 BALEEN: STOCHASTIC SELECTION VIA INFORMATION BOTTLENECK

In this section, we present our approach to improving SSMs via information bottleneck. In Sec. 3.1, we introduce the state space information bottleneck framework, designed to enhance the context compression capability, generalization, and interpretability of SSMs. In Sec. 3.2, we propose two novel SSM architectures that incorporates principled randomness modeling within the SSIB framework. Finally, in Sec. 3.3, we demonstrate how our architectures is equipped with inherent interpretability for understanding the behavior of SSM-based LLMs.

3.1 STATE SPACE INFORMATION BOTTLENECK

We focus on sequence-to-sequence modeling tasks using the SSM architecture. Let $\mathcal{X}, \mathcal{Y} \subseteq \mathbb{R}^{T \times D}$ represent the domains of input and target sequences, respectively. The training set consists of IID samples from the unknown joint distribution $\mathbb{P}(X, Y) = \mathbb{P}(Y|X)\mathbb{P}(X)^1$, which is supported

¹When referring to a probability density or mass function (PDF or PMF), we will omit the subscripts specifying the random variables it is defined over, as long as they are clear from the context according to the PDF or PMF's arguments or parameters.

on $\mathcal{X}$ and $\mathcal{Y}$. In the case of causal language modeling (Radford et al., 2018; 2019; Brown et al., 2020), $\mathbb{P}(\mathbf{Y}|\mathbf{X}) = \delta_{\mathbf{X}^*}(\mathbf{Y})$, where $\mathbf{X}^* = [\mathbf{x}_2^\top, \dots, \mathbf{x}_T^\top, \text{EOS}]^\top$ is a shifted version of $\mathbf{X}$ with EOS (special token) padding.

We aim to train a sequence-to-sequence SSM following Eq. 1: $\text{SSM}_{\Theta, \Phi} = g_{\Theta} \circ h_{\Phi} : \mathcal{X} \rightarrow \mathcal{Y}$, where *contextual encoder* h_{Φ} transforms subsequences of inputs $\mathbf{X}$ into the memory states $\mathbf{H}$ with parameters $\Phi = \{\mathbf{A}, \mathbf{W}_B, \mathbf{W}_{\Delta}\}$, and *predictor* g_{Θ} decodes the memory states $\mathbf{H}$ into the predicted sequence $\hat{\mathbf{Y}} \in \mathcal{Y}$ with $\Theta = \{\mathbf{W}_C\}$ (see Eq. 1)². An intrinsic property of SSMs is that $\mathbf{H}_t$ depends on $\mathbf{X}$ exclusively through its preceding tokens $\mathbf{X}_{\leq t} = [\mathbf{x}_1^\top, \dots, \mathbf{x}_t^\top]^\top$. And the prediction $\hat{\mathbf{y}}_t$ is conditionally independent of other variables given $\mathbf{H}_t$.

Naively applying the information bottleneck to SSMs presents unique challenges, as sequential data exhibit complex long-range dependencies that go beyond the IID assumptions (Alemi et al., 2016; Poole et al., 2019) or the local-dependence hypothesis (Wu et al., 2020; Miao et al., 2022). Based on the Markov property, a surrogate objective that upper-bounds the original objective can be derived as follows:

$$\min_{\Theta, \Phi} \underbrace{-\mathbb{E}_{\mathbf{H}, \mathbf{Y}} \left[\sum_{t=1}^T \log \mathbb{P}^{\Theta}(\mathbf{y}_t | \mathbf{H}_t) \right]}_{L_{\text{CE}}} + \beta \underbrace{\mathbb{E}_{\mathbf{X}} \left[\sum_{t=1}^T D_{\text{KL}}(\mathbb{P}^{\Phi}(\mathbf{H}_t | \mathbf{H}_{t-1}, \mathbf{X}_{\leq t}) \| \mathbb{Q}(\mathbf{H}_t | \mathbf{H}_{t-1})) \right]}_{L_{\text{KL}}} \quad (5)$$

Detailed derivations are deferred to Appendix A.2. We refer to the framework with learning objective in Eq. 5 as *State Space Information Bottleneck (SSIB)*.

Information-Theoretic Pitfalls of Mamba. Now we can revisit the training process of Mamba, revealing its equivalence to state space information bottleneck with $\beta = 0$, thus limited in compact selection. The training objective of Mamba is typically Maximum Likelihood Estimation (MLE): $\max_{\Theta, \Phi} \mathbb{E}_{\mathbf{X}, \mathbf{Y}} [\sum_{t=1}^{T-1} \log \mathbb{P}^{\Theta, \Phi}(\mathbf{y}_t | \mathbf{X}_{\leq t})]$. We argue that simply maximizing the mutual information between $\mathbf{y}_t$ and $\mathbf{H}_t$ only ensures $\mathbf{H}_t$ retains sufficient information for predicting $\mathbf{y}_t$. However, this does not guarantee that $\mathbf{H}_t$ achieves the maximal compression of $\mathbf{X}_{\leq t}$ and only focuses on tokens useful for prediction, potentially leading to information loss as the context length grows larger. Moreover, $\mathbf{H}_t$ may capture *spurious correlations* between $\mathbf{X}_{\leq t}$ and $\mathbf{y}_t$, making the predictions highly susceptible to noise (Chen et al., 2018; Wang et al., 2024). Instead, our approach explicitly models the stochastic relationship between $\mathbf{X}$ and $\mathbf{H}$, enabling the denoising of spurious patterns in $\mathbf{H}$ —a key factor for effective context compression, generalization, and interpretability.

3.2 BALEEN ARCHITECTURE AND LEARNING OBJECTIVE

In this subsection, we introduce the **Baleen** model family, which differs in *how random variables are modeled under the SSIB framework*. We begin by revisiting Eq. 1: the first term governs the transition of past memory, while the second term encodes the new token into the hidden memory state space. Since the state transition (i.e., $\mathbf{A}_t$) is the key component to randomize, whether randomness should also be incorporated into the new token encoding (i.e., $\mathbf{B}_t$) remains an open question. To investigate this, we first inject randomness solely into the state transition matrix, and then extend it to both the state transition and the new token encoding via the timestep variable Δ_t . The overall architectural design of SSIB is illustrated in Fig. 1.

Bernoulli-Distributed Transition. As mentioned before, each element in the memory state $\mathbf{H}_t$ represents a distinct component of past context (Gu et al., 2020), where a state transition value of one or zero determines whether the component is kept or removed from memory. It is naturally to model $\mathbf{A}_t$ as a variable follows Bernoulli distribution. In this case, $\mathbf{H}_t$ is an affine transformation of $\mathbf{A}_t$ and also follows a Bernoulli distribution. Moreover, each outcome of $\mathbf{H}_t$ corresponds to a unique outcome of $\mathbf{A}_t$ with the same probability. Therefore, the KL divergence is preserved under this affine transformation, i.e., the following equation holds:

$$D_{\text{KL}}(\mathbb{P}(\mathbf{H}_t | \mathbf{H}_{t-1}, \mathbf{X}_{\leq t}) \| \mathbb{Q}(\mathbf{H}_t | \mathbf{H}_{t-1})) = D_{\text{KL}}(\mathbb{P}(\mathbf{A}_t) \| \mathbb{Q}(\mathbf{A}_t)) \quad (6)$$

²Without loss of generality, we only consider SSMs with a single layer, without channel-mixing layer (Fu et al., 2022). We omit parameters for the embedding layer and the output head.

where $\mathbb{Q}(\mathbf{H}|\mathbf{H}_{t-1})$ is the corresponding affine transformation of $\mathbb{Q}(\mathbf{A}_t)$.

Because variational bound Eq. 5 holds true for any prior distribution $\mathbb{Q}$, we can simply define the prior distribution of $\mathbf{A}_t$ as $\text{Bern}(p_t \mathbf{1}_N \mathbf{1}_D^\top)$ where $p_t \in [0, 1]$ is a hyper-parameter. In training process, we implement the sampling process using the gumbel-softmax technique to ensure differentiability (Bengio et al., 2013; Jang et al., 2016; Maddison et al., 2016). Then the KL loss in Eq. 5 is simplified as follows:

$$L_{\text{KL}} = \mathbb{E} \left[\sum_{t,n,d} \mathbf{A}_t^{(n,d)} \log \frac{\mathbf{A}_t^{(n,d)}}{p_t} + (1 - \mathbf{A}_t^{(n,d)}) \log \frac{1 - \mathbf{A}_t^{(n,d)}}{1 - p_t} \right] \quad (7)$$

where the summation is taken over $t \in [T], n \in [N], d \in [D]$. Detailed derivations can be found in Appendix A.3.

Exponential-Distributed Timestep. To simultaneously inject randomness into both $\mathbf{A}_t$ and $\mathbf{B}_t$ is challenging, especially $\mathbf{A}$ and $\mathbf{B}_t$ should conform to different distributions even discrete and continuous are not the same. Based on our observation on both $\mathbf{A}$ and $\mathbf{B}_t$ are parameterized with Δ_t , we may assume Δ_t a random variable to implicitly affect both terms. In SSMs, Δ_t is often interpreted as the discretization timestep from the perspective of a dynamic system (Gu & Dao, 2023), while the exponential distribution is classically used to model the time interval. Motivated by this connection, we propose to redefine Δ_t in Eq. 2a as a random variable following Exponential distribution. In this case, $\mathbf{H}_t$ has an explicit expression form:

$$\mathbf{H}_t = \mathbf{H}_{t-1} \exp(\hat{\mathbf{A}}_t \text{diag}(\Delta_t)) + \mathbf{1}_N \mathbf{x}_t \hat{\mathbf{B}}_t \Delta_t \quad (8)$$

where the first term follows Pareto distribution and the second term still follows Exponential distribution. their sum does not belong to any well-known class of distributions. Since $\mathbf{H}_t$ as a function of Δ_t is not invertible, the KL divergence in Eq. 5 cannot be directly reduced to that between Δ_t and its prior like Eq. 6 shows. Nevertheless, we can show that this KL divergence serves as an upper bound of the original one, which allows us to directly optimize with respect to Δ_t . Further details are provided in Appendix A.3.

We then choose the prior distribution $\mathbb{Q}(\Delta_t)$ as an Exponential distribution, $\text{Exp}(\lambda_t \mathbf{1}_D)$, which yields an upper bound on the KL divergence in Eq. 5, as shown below:

$$L_{\text{KL}} = \mathbb{E} \left[\sum_{t,d} \log \frac{\Delta_t^{(d)}}{\lambda_t} - (\Delta_t^{(d)} - \lambda_t) \frac{1}{\Delta_t^{(d)}} \right], \quad (9)$$

where the summation is taken over $t \in [T], d \in [D]$.

We name the two novel architectures with stochastic selection *Baleen-B* (Bernoulli) and *Baleen-E* (Exponential). While Baleen-B polarizes the state transition values to ensure a compact representation, Baleen-E implements a *trade-off* between incorporating more past memory and more new encoded token information. This behavior can be explained as follows: when Δ_t tends to infinity, the first term in Eq. 8 vanishes due to the negative $\hat{\mathbf{A}}_t$, and $\mathbf{H}_t$ depends solely on the new token embeddings. Conversely, when Δ_t tends to zero the second term vanishes, and $\mathbf{H}_t$ is entirely inherited from past memory. Therefore, Δ_t effectively selects information from the new input token while simultaneously clearing past memory to make room for it. Note that our framework not only supports pre-training but also enables fine-tuning from a pre-trained SSM. This is achieved without introducing any additional modules—only by converting the deterministic gates into stochastic ones in a plug-and-play manner.

3.3 INHERENT INTERPRETABILITY

As previously discussed, each element in the state transition matrix $\mathbf{A}_t$ governs whether a specific component of the historical context is retained or discarded in memory (Gu et al., 2020). Preserving a larger subset of components enables the current input token to dynamically interact with and update the states of these retained components, thereby encoding critical information into the model’s memory. This mechanism provides inherent interpretability to the Baleen architecture.

$$\text{Importance}(t) = \frac{1}{ND} \sum_{d=1}^D \sum_{n=1}^N \mathbf{A}_t^{(n,d)}, \quad t \in [T] \quad (10)$$

Table 1: Accuracy comparison on benchmark datasets. Best results are in bold, second best underlined.

Method	ARC-C	ARC-E	BoolQ	GPQA	Hella.	MMLU	OBQA	PIQA	SIQA	TruthfulQA	Wino.	Avg.
Pretrained 370M models												
Baleen-B	18.94	36.20	<u>59.39</u>	24.33	<u>28.02</u>	22.97	14.80	59.14	32.91	23.38	51.22	<u>33.75</u>
Baleen-E	17.58	39.35	60.52	24.11	28.18	23.01	14.20	60.45	33.11	23.62	48.38	33.86
Gated DeltaNet	23.29	32.45	51.25	24.11	26.99	23.29	15.00	56.96	35.11	<u>25.21</u>	51.85	33.23
Mamba2	17.83	<u>37.54</u>	49.66	24.33	27.84	22.92	14.40	<u>59.96</u>	33.01	23.87	<u>50.67</u>	32.91
Mamba	<u>22.01</u>	29.25	45.69	25.89	25.91	24.98	13.20	54.62	32.70	<u>25.21</u>	49.49	31.72
RetNet	21.67	30.85	57.13	24.33	26.50	23.60	14.20	55.93	<u>33.93</u>	24.11	48.78	32.82
RWKV6	21.76	31.06	46.21	<u>25.67</u>	26.92	<u>24.10</u>	15.80	55.82	33.01	26.56	49.96	32.44
GLA	21.93	30.26	44.56	24.78	26.52	23.76	15.60	56.42	33.62	25.09	49.41	32.00
Finetuned 7B models												
Baleen-B 7B	31.23	61.99	73.06	27.23	39.37	34.76	21.80	67.57	32.91	27.66	58.01	43.24
Baleen-E 7B	31.40	60.94	71.74	20.54	40.26	33.70	22.40	68.28	32.96	27.54	56.99	42.43
Mamba2 7B	33.19	62.08	62.63	27.01	38.06	32.84	19.80	67.63	32.91	27.17	57.14	41.86

To operationalize interpretation during inference, we first compute the expectation of the stochastic transition matrices A_t and use it as state transition matrix in Eq. 1. We then derive token-level importance scores by averaging the magnitudes of the transition matrices across different components (i.e., averaging over embedding dimension D and state dimension N as Eq. 10 shows). These scores enable the identification of salient tokens through a top- k ranking strategy, offering transparent insights into the model’s decision-making process during prediction.

4 EXPERIMENTS

In this section, we will validate our effectiveness in common language benchmarks, providing faithful inherent interpretation, and ensuring robustness under perturbations.

Settings. We pre-trained 370M Baleen-B and Baleen-E models based on Mamba2 (Dao & Gu, 2024) on 20B tokens from SlimPajama dataset (Soboleva et al., 2023), and fine-tuned 7B Baleen-B and Baleen-E models from a pre-trained Mamba2 model (Codestral (AI@Mistral, 2024)) on 0.5B instruction-following formatted tokens from Crystal dataset (Liu et al., 2023). We mainly compare our method with the vallina Mamba2 model, and add other state space or linear attention models as baselines including RetNet (Sun et al., 2023), RWKV6 (Peng et al., 2023), GLA (Yang et al., 2023) Mamba (Gu & Dao, 2023), and recently proposed Gated DeltaNet (Yang et al., 2024a).

4.1 LANGUAGE BENCHMARKS

Pre-trained 370M Models. Baleen-B (Baleen-B) and Baleen-E (Baleen-E) achieve the best overall performance, with average accuracies of **33.75%** and **33.86%**, both surpassing all baselines outperforming all baselines including the strong Gated-DeltaNet (33.23%), which is the architecture behind the popular industry model Qwen3-Next (Qwen-Team, 2025). On commonsense reasoning tasks, Baleen-E leads ARC-Easy (**39.35%**, +1.81 over Mamba2) and PIQA (**60.45%**, +0.49), while Baleen-B ranks first on Winogrande (**51.22%**, +0.55). For language understanding, their MMLU (22.97% and 23.01%) scores slightly exceed Mamba2 (22.92%). Overall, both Baleen variants show consistent gains over Mamba2, confirming the benefit of our design on top of the Mamba2 architecture.

Fine-tuned 7B Models. After instruction tuning, Baleen-B attains the highest average accuracy of **43.24%**, followed by Baleen-E (**42.43%**), both ahead of Mamba2 (41.86%). In commonsense reasoning, Baleen-B excels on BoolQ (**73.06%**, +10.43), while Baleen-E leads PIQA (**68.28%**, +1.65). Both outperform Mamba2 on Winogrande (58.01% and 56.99% vs. 57.14%). For language understanding, Baleen-B reaches 34.76% on MMLU (+1.92), while Baleen-E achieves the highest HellaSwag (40.26%, +2.2) and competitive MMLU (33.70%) scores. On TruthfulQA, Baleen-B achieves 27.66% (+0.49), slightly above Mamba2, while Baleen-E performs slightly below Mamba2. Overall, Baleen-B and Baleen-E consistently outperform the Mamba2 baseline, delivering stronger commonsense reasoning and steady gains in language understanding and truthfulness.

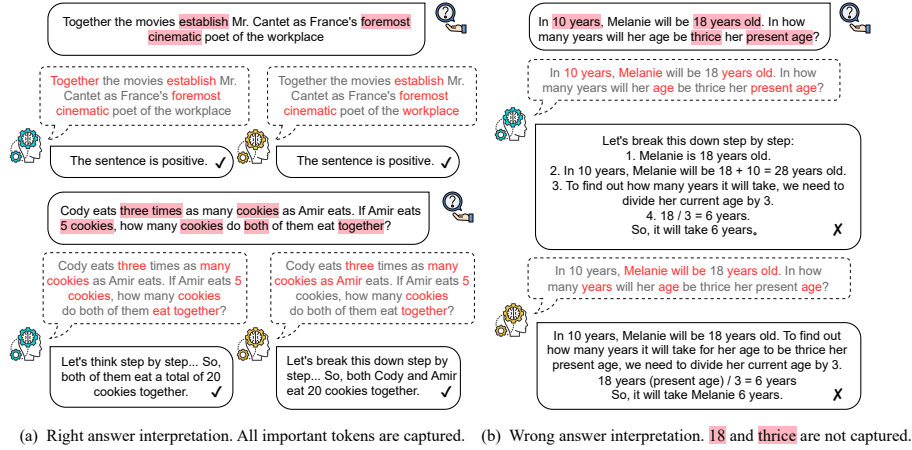


Figure 2: Interpretation Examples on SST and GSM8K. Human annotated important tokens in questions are highlighted in red background, and model interpretation after top-k ranking ($k = 4/8$ for SST/GSM8K) are highlighted in red font. The blue avatar refers to Baleen-E 7B and yellow avatar refers to Baleen-B 7B.

4.2 INHERENT INTERPRETATION

Qualitative Example We test fine-tuned 7B model in SST and GSM8K to yield inherent interpretation when generating answer. We visualize three samples with human annotated ground truth interpretation from previous literature (Wu et al., 2023) in Fig. 2.

As Fig. 2 (a) shows, both Baleen-E (blue) and Baleen-B (yellow) are able to provide accurate interpretation aligned with human annotations well when generating correct answers. For sentiment analysis, three out of the four most important tokens indicating positive emotion are identified. For the math problem, key elements including objects, numbers, and the word “together” are selected.

In contrast, Fig. 2 (b) illustrates how the model makes a wrong answer in a math problem: It fails to properly attend to the tokens “18” and “thrice” in the input. Consequently, the model incorrectly treats Melanie’s current age as “18” instead of “8”, and divides the age by 3 rather than multiplying it by 3. These two mistakes are directly attributable to the model’s inattention to “18” and “thrice”, which clearly explains why the output is wrong. Such analysis is useful for monitoring the model’s long-form text generation, enabling early identification of errors and preventing wasted computation. Moreover, it reveals potential weaknesses in the model and provides guidance for further improvement.

Quantitative Evaluation. We conduct quantitative evaluation regarding model interpretations and report results in Tab. 2. Fidelity AUC (Zhu et al., 2024; Jafari et al., 2024) is often adopted to reduce the evaluation bias of single sparsity value (i.e., Fidelity@k) which cannot provide consistent evaluations. We use four well-established text classification tasks SNLI (Bowman et al., 2015), IMDB (Maas et al., 2011), SST2 (Socher et al., 2013) and Rotten Tomatoes (Pang & Lee, 2005), and fine-tuned pretrained 370M model on them. We compare our methods (self-interpretation) with commonly used gradient-based post-hoc methods including Gradient $\times$ Input (Shrikumar et al., 2017), Integrated Gradients (IG) (Sundararajan et al., 2017) and GradCAM (Selvaraju et al., 2017). For the fair comparison, we ensure that the interpreted model remains the same; thus, Baleen-B 370M and Baleen-E 370M serve as the base models. Notably, our method is self-interpretable and can generate interpretations without additional compute budget.

Table 2: Explanation performance on two base models trained with Baleen-E and Baleen-B respectively. “RT.” refers to Rotten Tomatoes.

Method	SNLI	IMDb	SST2	RT.
Baleen-B				
IG	44.00	40.21	39.94	40.50
GradX	48.85	42.98	47.96	40.94
GradCAM	54.98	49.59	42.57	42.00
Self (Ours)	55.81	45.24	57.65	51.49
Baleen-E				
IG	24.23	2.45	38.47	2.27
GradX	48.88	43.69	47.79	41.63
GradCAM	54.73	52.69	39.19	48.65
Self (Ours)	54.31	51.66	62.50	62.17

As demonstrated in Tab. 2, both Baleen-B and Baleen-E consistently generate highly faithful interpretations compared to existing methods, with their average performance significantly surpassing

Table 3: Adversarial attack results of CIFAR-10 image classification on sequences of pixels.

Models	Perturbation region (seq. length = 1024)						Avg.
	[0:0]	[0:32]	[0:96]	[928:1024]	[992:1024]		
RWKV	0.474	0.150 (↓ 68.35%)	0.466 (↓ 1.58%)	0.138 (↓ 70.88%)	0.460 (↓ 2.91%)		0.338±0.177
GLA	0.631	0.623 (↓ 1.42%)	0.597 (↓ 5.41%)	<u>0.332</u> (↓ 47.36%)	0.496 (↓ 21.42%)		0.536±0.126
Mamba	0.672	0.655 (↓ 2.54%)	0.631 (↓ 6.19%)	0.102 (↓ 84.77%)	0.152 (↓ 77.35%)		0.442±0.289
Mamba2	0.673	0.495 (↓ 26.51%)	0.470 (↓ 30.17%)	0.106 (↓ 84.17%)	0.183 (↓ 72.74%)		0.385±0.235
Baleen-B	<u>0.659</u>	0.634 (↓ 3.82%)	0.603 (↓ 8.58%)	0.314 (↓ 52.37%)	0.453 (↓ 31.23%)		<u>0.532</u> ±0.146
Baleen-E	0.643	<u>0.639</u> (↓ 0.57%)	<u>0.628</u> (↓ 2.33%)	0.365 (↓ 43.16%)	0.433 (↓ 32.63%)		0.542 ± 0.132

baseline approaches. This inherent self-interpretability enables real-time monitoring of model behavior, allowing practitioners to promptly detect errors during inference rather than relying on post hoc analysis.

4.3 ADVERSARIAL ROBUSTNESS

Position Bias Attack. SSMs exhibit position bias, which can be utilized to compromise their performance (Wang et al., 2024). To assess whether our method mitigates this bias, we evaluate model robustness against corrupted inputs through *image classification on sequences of pixels* (Tay et al., 2020). For this task, each $W \times H$ image is flattened into a sequence of WH RGB pixel tokens, which are then embedded into the hidden state dimension and processed by sequence modeling blocks. Following ViT (Dosovitskiy et al., 2021), we append a learnable class token at the end of the sequence. This token is subsequently mapped to logits via a classifier head for image classification. This design introduces a vulnerability: since predictions rely on the class token, position bias implies that tokens closer to it exert a greater influence on the output. Consequently, SSMs with weaker robustness are particularly vulnerable to perturbations in trailing tokens near the class token. To highlight this bias, we generate corrupted inputs by perturbing both leading and trailing tokens.

In our image classification experiments on CIFAR-10 (Tab. 3), we find both our variants demonstrate balanced sensitivity to perturbations at leading and trailing positions. Under leading-token corruption ($[0:32]$), Baleen-B and Baleen-E retain 0.634 accuracy (3.8% drop) and 0.639 (0.6% drop), respectively, maintaining moderate performance degradation comparable to the original Mamba while outperforming the base model Mamba2. However, trailing-token corruption ($[992:1024]$) leads to catastrophic accuracy losses for both Mamba and Mamba2 ($>70\%$). In contrast, SSIBs demonstrate significantly improved robustness, limiting accuracy drops to approximately 30% under the same conditions. Notably, Baleen-B achieves the highest average accuracy across all compared methods.

Prompt Attack. To further evaluate the robustness improvement by our method, we conduct prompt attack experiments on our fine-tuned 7B language models using WNLI and SST2 dataset (Wang et al., 2018). Following the setup in PromptBench (Zhu et al., 2023a), we apply four different types of perturbation: DeepWordBug, TextBugger, CheckList, and StressTest (Zhu et al., 2023b) varying from character-level attack to sentence-level attack. Robustness is assessed by measuring the performance drop ratio (PDR) in prediction accuracy between normal and perturbed prompts.

Tab. 4 compares the performance degradation of our fine-tuned 7B Baleen-E and Baleen-B models against 7B Mamba2. As shown, both Baleen-E and Baleen-B consistently achieve lower drop ratios across most adversarial settings, indicating stronger robustness to perturbations. In particular, Baleen-B demonstrates superior resilience on SST2 under Checklist and StressTest, with performance drop ratios that are 44.66% and 18.55% lower than those of Mamba2, respectively. Meanwhile, Baleen-E shows competitive robustness under DeepWordBug and TextBugger, achieving drop ratios that are 32.14% and 18.16% lower than Mamba2. On WNLI, Baleen-E substantially outperforms the other models on Checklist and TextBugger, achieving drop ratios as low as 17.65 and 0.00 respectively. These results highlight that our fine-tuned Baleen models not only mitigate performance degradation more effectively than Mamba2 but also complement each other in handling different types of adversarial attacks.

5 RELATED WORK

Linear State Space Modeling as Online Learning. Previous works (Sun et al., 2024; Liu et al., 2024; Behrouz et al., 2024) have proposed to unify the computational structure of linear SSMs from

Table 4: Performance drop ratio (lower is better) results after adversarial attack on PromptBench.

Model	SST2				WNLI			
	Checklist	DeepWordBug	StressTest	TextBugger	Checklist	DeepWordBug	StressTest	TextBugger
Baleen-B	32.26	43.16	43.48	71.58	61.76	6.67	94.74	9.38
Baleen-E	39.08	38.37	71.26	70.45	17.65	14.71	87.50	0.00
Mamba2	76.92	70.51	62.03	88.61	80.00	13.33	100.00	10.00

an online learning perspective. The update rule of SSMs can be generally understood as doing a single step of online stochastic gradient descent, with respect to the state spaces $\mathbf{H}_t$, for optimizing an objective $L(\mathbf{H}_t) = \|\mathbf{H}_t - \alpha_t \mathbf{H}_{t-1}\|$ at the time stamp t to control the magnitude of the state update. These online learning objectives often introduce additional regularization terms for better modeling the key-value associations, resulting in different variants of state updates consisting of gating (Sun et al., 2023; Gu & Dao, 2023; Yang et al., 2023; Peng et al., 2023; De et al., 2024; Goldstein et al., 2024), delta rule (Schlag et al., 2021; Yang et al., 2024b;a) and momentum (Behrouz et al., 2024). Our approach is loosely connected to this perspective, in the sense that we regard the hidden states as an information bottleneck between the input and the target sequences, and the KL term functions as a regularizer of the hidden state $\mathbf{H}_t$ with respect to a prior distribution $\mathbb{Q}$. Since Linear Attention (Qin et al., 2022; Sun et al., 2023) can be viewed as SSMs with linear recurrent updates of two-dimensional associative memories, our approach can be applied to a broader spectrum of architectures (MiniMax et al., 2025; Ren et al., 2023; 2024; Lieber et al., 2024; Ma et al., 2022; 2024) consisting of linear recurrent models.

Interpretation Methods. Previous interpretation methods can form into two categories (Arrieta et al., 2020; Rudin, 2019). The first category, known as post-hoc methods, operates on already-trained models. For instance, Gradient $\times$ Input (Shrikumar et al., 2017) compute the product of the input and the gradients w.r.t. the input as the importance scores. GradCAM (Selvaraju et al., 2017) generalize the operations to intermediate embeddings and the corresponding gradients. The second category comprises self-interpretable models. These models integrate interpretable modules into model architectures, rooted in principled mechanism such as causality (Wu et al., 2022), information bottleneck (Jiang et al., 2020) or and even attention mechanism (Smilkov et al., 2017). These models are specifically designed to extract and utilize minimal significant information during model training. For instance, vanilla attention score used to be treat as self-interpretation, but are proved to be not faithful (Jain & Wallace, 2019), and then improved via modified attention mechanism (Chrysostomou & Aletras, 2021) and interpretation deduction (Liu et al., 2022). Our method belongs to self-interpretable methods with theoretical principle.

Adversarial Robustness. As large language models (LLMs) increasingly dominate natural language processing and permeate daily life through their conversational capabilities, concerns about their reliability and alignment with human values have grown. Robustness is a key factor, requiring models to remain reliable even under adversarial or challenging conditions. Existing approaches mostly rely on post-hoc defenses such as detectors or input/output filters (Dong et al., 2024), but more subtle attacks (e.g., noisy data injection during training) highlight the need for inherently robust architectures. Prior studies explore robustness enhancements through mutual information regularization (Wang et al.), information bottleneck methods (Zhang et al., 2022), or selective fine-tuning (Kim et al., 2023), though these works mainly target smaller models like BERT or RoBERTa. To the best of our knowledge, we are the first to demonstrate robustness at the scale of modern LLMs (e.g., 7B parameters) using a scalable SSM-based architecture.

6 CONCLUSION

In this work, we introduce *Baleen*, a family of SSM architectures that explicitly maximizes context compression with reliable interpretation based on *State Space Information Bottleneck* framework. It is realized by stochastic modeling of the selection mechanism, enforces memory states to be *minimal and sufficient* representation of the past context. Extensive experiments show *Baleen* addresses the limitations of existing SSMs in selectivity, robustness, and interpretability.

REFERENCES

- AI@Mistral. Mamba-codestral-7b-v0.1: A 7b parameter state-space model for code generation. <https://huggingface.co/mistralai/Mamba-Codestral-7B-v0.1>, 2024. Accessed: [Insert Access Date].
- Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.
- Ameen Ali, Itamar Zimmerman, and Lior Wolf. The hidden attention of mamba models. *arXiv preprint arXiv:2403.01590*, 2024.
- Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Zou, Atri Rudra, and Christopher Ré. Zoology: Measuring and improving recall in efficient language models. *arXiv preprint arXiv:2312.04927*, 2023.
- Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bannetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020.
- Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. *arXiv preprint arXiv: 2501.00663*, 2024. URL <https://arxiv.org/abs/2501.00663v1>.
- Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.
- Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In *International conference on machine learning*, pp. 883–892. PMLR, 2018.
- George Chrysostomou and Nikolaos Aletras. Improving the faithfulness of attention-based explanations with task-specific information for text classification. *arXiv preprint arXiv:2105.02657*, 2021.
- Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.
- Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024.
- Soham De, Samuel L Smith, Anushan Fernando, Aleksandar Botev, George Cristian-Muraru, Albert Gu, Ruba Haroun, Leonard Berrada, Yutian Chen, Srivatsan Srinivasan, et al. Griffin: Mixing gated linear recurrences with local attention for efficient language models. *arXiv preprint arXiv:2402.19427*, 2024.
- Zhichen Dong, Zhanhui Zhou, Chao Yang, Jing Shao, and Yu Qiao. Attacks, defenses and evaluations for llm conversation safety: A survey. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6734–6747, 2024.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proceedings of ICLR*, 2021.
- Daniel Y Fu, Tri Dao, Khaled K Saab, Armin W Thomas, Atri Rudra, and Christopher Ré. Hungry hungry hippos: Towards language modeling with state space models. *arXiv preprint arXiv:2212.14052*, 2022.

- Daniel Goldstein, Fares Obeid, Eric Alcaide, Guangyu Song, and Eugene Cheah. Goldfinch: High performance rwkv/transformer hybrid with linear pre-fill and extreme kv-cache compression. *arXiv preprint arXiv: 2407.12077*, 2024. URL <https://arxiv.org/abs/2407.12077v1>.
- Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.
- Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. Hippo: Recurrent memory with optimal polynomial projections. *Advances in neural information processing systems*, 33: 1474–1487, 2020.
- Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021a.
- Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34:572–585, 2021b.
- Albert Gu, Karan Goel, Ankit Gupta, and Christopher Ré. On the parameterization and initialization of diagonal state space models. *Advances in Neural Information Processing Systems*, 35:35971–35983, 2022a.
- Albert Gu, Isys Johnson, Aman Timalina, Atri Rudra, and Christopher Ré. How to train your hippo: State space models with generalized orthogonal basis projections. *arXiv preprint arXiv:2206.12037*, 2022b.
- Ankit Gupta, Albert Gu, and Jonathan Berant. Diagonal state spaces are as effective as structured state spaces. *Advances in Neural Information Processing Systems*, 35:22982–22994, 2022.
- Farnoush Rezaei Jafari, Grégoire Montavon, Klaus-Robert Müller, and Oliver Eberle. Mambalrp: Explaining selective state space sequence models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- Sarthak Jain and Byron C Wallace. Attention is not explanation. *arXiv preprint arXiv:1902.10186*, 2019.
- Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.
- Zhiying Jiang, Raphael Tang, Ji Xin, and Jimmy Lin. Inserting information bottlenecks for attribution in transformers. *arXiv preprint arXiv:2012.13838*, 2020.
- Jaehyung Kim, Yuning Mao, Rui Hou, Hanchao Yu, Davis Liang, Pascale Fung, Qifan Wang, Fuli Feng, Lifu Huang, and Madian Khabisa. Roast: Robustifying language models via adversarial perturbation with selective training. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 3412–3444, 2023.
- Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, Omri Abend, Raz Alon, Tomer Asida, Amir Bergman, Roman Glozman, Michael Gokhman, Avashalom Manevich, Nir Ratner, Noam Rozen, Erez Shwartz, Mor Zusman, and Yoav Shoham. Jamba: A hybrid transformer-mamba language model. *arXiv preprint arXiv: 2403.19887*, 2024. URL <https://arxiv.org/abs/2403.19887v1>.
- Bo Liu, Rui Wang, Lemeng Wu, Yihao Feng, Peter Stone, and Qiang Liu. Longhorn: State space models are amortized online learners. *arXiv preprint arXiv: 2407.14207*, 2024. URL <https://arxiv.org/abs/2407.14207v4>.
- Yibing Liu, Haoliang Li, Yangyang Guo, Chenqi Kong, Jing Li, and Shiqi Wang. Rethinking attention-model explainability through faithfulness violation test. In *International Conference on Machine Learning*, pp. 13807–13824. PMLR, 2022.

- Zhengzhong Liu, Aurick Qiao, Willie Neiswanger, Hongyi Wang, Bowen Tan, Tianhua Tao, Junbo Li, Yuqi Wang, Suqi Sun, Omkar Pangarkar, et al. Llm360: Towards fully transparent open-source llms. *arXiv preprint arXiv:2312.06550*, 2023.
- Xuezhe Ma, Chunting Zhou, Xiang Kong, Junxian He, Liangke Gui, Graham Neubig, Jonathan May, and Luke Zettlemoyer. Mega: moving average equipped gated attention. *arXiv preprint arXiv:2209.10655*, 2022.
- Xuezhe Ma, Xiaomeng Yang, Wenhan Xiong, Beidi Chen, Lili Yu, Hao Zhang, Jonathan May, Luke Zettlemoyer, Omer Levy, and Chunting Zhou. Megalodon: Efficient llm pretraining and inference with unlimited context length. *arXiv preprint arXiv:2404.08801*, 2024.
- 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.
- Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.
- Siqi Miao, Mia Liu, and Pan Li. Interpretable and generalizable graph learning via stochastic attention mechanism. In *International Conference on Machine Learning*, pp. 15524–15543. PMLR, 2022.
- MiniMax, Aonian Li, Bangwei Gong, Bo Yang, Boji Shan, Chang Liu, Cheng Zhu, Chunhao Zhang, Congchao Guo, Da Chen, Dong Li, Enwei Jiao, Gengxin Li, Guojun Zhang, Haohai Sun, Houze Dong, Jiadai Zhu, Jiaqi Zhuang, Jiayuan Song, Jin Zhu, Jingtao Han, Jingyang Li, Junbin Xie, Junhao Xu, Junjie Yan, Kaishun Zhang, Kecheng Xiao, Kexi Kang, Le Han, Leyang Wang, Lianfei Yu, Liheng Feng, Lin Zheng, Linbo Chai, Long Xing, Meizhi Ju, Mingyuan Chi, Mozhi Zhang, Peikai Huang, Pengcheng Niu, Pengfei Li, Pengyu Zhao, Qi Yang, Qidi Xu, Qiexiang Wang, Qin Wang, Qihui Li, Ruitao Leng, Shengmin Shi, Shuqi Yu, Sichen Li, Songquan Zhu, Tao Huang, Tianrun Liang, Weigao Sun, Weixuan Sun, Weiyu Cheng, Wenkai Li, Xiangjun Song, Xiao Su, Xiaodong Han, Xinjie Zhang, Xinzhu Hou, Xu Min, Xun Zou, Xuyang Shen, Yan Gong, Yingjie Zhu, Yipeng Zhou, Yiran Zhong, Yongyi Hu, Yuanxiang Fan, Yue Yu, Yufeng Yang, Yuhao Li, Yunan Huang, Yunji Li, Yunpeng Huang, Yunzhi Xu, Yuxin Mao, Zehan Li, Zekang Li, Zewei Tao, Zewen Ying, Zhaoyang Cong, Zhen Qin, Zhenhua Fan, Zhihang Yu, Zhuo Jiang, and Zijia Wu. Minimax-01: Scaling foundation models with lightning attention. *arXiv preprint arXiv:2501.08313*, 2025. URL <https://arxiv.org/abs/2501.08313v1>.
- Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *arXiv preprint cs/0506075*, 2005.
- Jongho Park, Jaeseung Park, Zheyang Xiong, Nayoung Lee, Jaewoong Cho, Samet Oymak, Kangwook Lee, and Dimitris Papailiopoulos. Can mamba learn how to learn? a comparative study on in-context learning tasks. *arXiv preprint arXiv:2402.04248*, 2024.
- Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, Kranthi Kiran GV, et al. Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*, 2023.
- Michael Poli, Armin W Thomas, Eric Nguyen, Pragaash Ponnusamy, Björn Deiseroth, Kristian Kersting, Taiji Suzuki, Brian Hie, Stefano Ermon, Christopher Ré, et al. Mechanistic design and scaling of hybrid architectures. *arXiv preprint arXiv:2403.17844*, 2024.
- Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pp. 5171–5180. PMLR, 2019.
- Zhen Qin, Xiaodong Han, Weixuan Sun, Dongxu Li, Lingpeng Kong, Nick Barnes, and Yiran Zhong. The devil in linear transformer. *Conference on Empirical Methods in Natural Language Processing*, 2022. doi: 10.48550/arXiv.2210.10340. URL <https://arxiv.org/abs/2210.10340v1>.
- Qwen-Team. Qwen3 technical report, 2025. URL <https://arxiv.org/abs/2505.09388>.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.
- Liliang Ren, Yang Liu, Shuohang Wang, Yichong Xu, Chenguang Zhu, and ChengXiang Zhai. Sparse modular activation for efficient sequence modeling. *NEURIPS*, 2023. URL <https://arxiv.org/abs/2306.11197v1>.
- Liliang Ren, Yang Liu, Yadong Lu, yelong shen, Chen Liang, and Weizhu Chen. Samba: Simple hybrid state space models for efficient unlimited context language modeling. 2024. URL <https://openreview.net/forum?id=okbDFg9W0n¬eId=okbDFg9W0n>.
- Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.
- Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9355–9366. PMLR, 2021. URL <http://proceedings.mlr.press/v139/schlag21a.html>.
- Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.
- Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pp. 3145–3153. PMIR, 2017.
- Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.
- Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. Slimpajama: A 627b token cleaned and deduplicated version of redpajama, 2023.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.
- Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, Tatsunori Hashimoto, and Carlos Guestrin. Learning to (learn at test time): Rnns with expressive hidden states. *arXiv preprint arXiv: 2407.04620*, 2024. URL <https://arxiv.org/abs/2407.04620v1>.
- Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*, 2023.
- Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.
- Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena: A benchmark for efficient transformers. *arXiv preprint arXiv:2011.04006*, 2020.
- Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All You Need. In *Proceedings of NeurIPS*, 2017.

- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, 2018.
- Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. Infobert: Improving robustness of language models from an information theoretic perspective. In *International Conference on Learning Representations*.
- Peihao Wang, Ruisi Cai, Yuehao Wang, Jiajun Zhu, Pragya Srivastava, Zhangyang Wang, and Pan Li. Understanding and mitigating bottlenecks of state space models through the lens of recency and over-smoothing. *arXiv preprint arXiv:2501.00658*, 2024.
- Skyler Wu, Eric Meng Shen, Charumathi Badrinath, Jiaqi Ma, and Himabindu Lakkaraju. Analyzing chain-of-thought prompting in large language models via gradient-based feature attributions. *arXiv preprint arXiv:2307.13339*, 2023.
- Tailin Wu, Hongyu Ren, Pan Li, and Jure Leskovec. Graph information bottleneck. *Advances in Neural Information Processing Systems*, 33:20437–20448, 2020.
- Ying-Xin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. Discovering invariant rationales for graph neural networks. *arXiv preprint arXiv:2201.12872*, 2022.
- Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. *arXiv preprint arXiv:2312.06635*, 2023.
- Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. *arXiv preprint arXiv: 2412.06464*, 2024a. URL <https://arxiv.org/abs/2412.06464v1>.
- Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. *arXiv preprint arXiv: 2406.06484*, 2024b. URL <https://arxiv.org/abs/2406.06484v1>.
- Cenyuan Zhang, Xiang Zhou, Yixin Wan, Xiaoqing Zheng, Kai-Wei Chang, and Cho-Jui Hsieh. Improving the adversarial robustness of nlp models by information bottleneck. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 3588–3598, 2022.
- Jiajun Zhu, Siqi Miao, Rex Ying, and Pan Li. Towards understanding sensitive and decisive patterns in explainable ai: A case study of model interpretation in geometric deep learning. *arXiv preprint arXiv:2407.00849*, 2024.
- Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Neil Zhenqiang Gong, Yue Zhang, et al. Promptbench: Towards evaluating the robustness of large language models on adversarial prompts. *arXiv preprint arXiv:2306.04528*, 2023a.
- Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Yue Zhang, Neil Gong, et al. Promptrobust: Towards evaluating the robustness of large language models on adversarial prompts. In *Proceedings of the 1st ACM Workshop on Large AI Systems and Models with Privacy and Safety Analysis*, pp. 57–68, 2023b.

A DEFERRED THEORY

In this section, we provide detailed derivations for SSIB framework with two types of stochasticity.

A.1 BACKGROUND

In this section, we provide readers with basic concepts in information theory. Below our introduction focuses on two random variables A, B . Extending the below definitions to random vectors/matrices is straightforward. Interested readers are referred to Cover (1999).

Entropy. The entropy of a random variable A quantifies how much information it carries in its distribution, often characterized by how many “bits” are needed to represent the random variable. It is defined as below:

$$\mathbb{H}(A) = -\mathbb{E}_A [\log \mathbb{P}(A)]. \quad (11)$$

Higher entropy indicates greater uncertainty in the possible values of A .

Mutual Information. The mutual information between two random variables A and B measures the amount of information A provides about B :

$$\mathbb{I}(A; B) = \mathbb{I}(B; A) = \mathbb{E}_{A,B} \left[\log \frac{\mathbb{P}(A, B)}{\mathbb{P}(A)\mathbb{P}(B)} \right]. \quad (12)$$

It captures the reduction in uncertainty of one variable given knowledge of the other.

KL Divergence. The Kullback-Leibler (KL) divergence measures how different a probability distribution $\mathbb{P}(A)$ is from a reference distribution $\mathbb{Q}(A)$:

$$D_{\text{KL}}(\mathbb{P}(A) \parallel \mathbb{Q}(A)) = \mathbb{E}_{A \sim \mathbb{P}} \left[\log \frac{\mathbb{P}(A)}{\mathbb{Q}(A)} \right]. \quad (13)$$

KL divergence is always non-negative and equals zero if and only if $\mathbb{P}(A) = \mathbb{Q}(A)$ almost everywhere.

As seen in Sec. 3, these information-theoretic metrics play a fundamental role in the information bottleneck framework. We assume all above quantities are well-defined throughout this paper. They are also closely related to each other. Below are some useful equalities among entropy, mutual information, and KL divergence:

$$\mathbb{H}(A|B) = \mathbb{H}(A, B) - \mathbb{H}(B) \quad (14)$$

$$\mathbb{I}(A; B) = \mathbb{H}(A) - \mathbb{H}(A|B) = \mathbb{H}(B) - \mathbb{H}(B|A) = \mathbb{H}(A) + \mathbb{H}(B) - \mathbb{H}(A, B), \quad (15)$$

$$\mathbb{I}(A; B) = D_{\text{KL}}(\mathbb{P}(A, B) \parallel \mathbb{P}(A)\mathbb{P}(B)). \quad (16)$$

A.2 SSIB TRAINING OBJECTIVE

In this section, we derive the main results in Sec. 3.1. We begin by specifying the PDFs for inputs $\mathbf{X}$, memory states $\mathbf{H}$, and ground-truth sequence $\mathbf{Y}$. First of all, $\mathbb{P}(\mathbf{X})$ or $\mathbb{P}_{\mathbf{X}}$ is an unknown distribution where input sequences are sampled from. The conditional distribution $\mathbb{P}(\mathbf{Y}|\mathbf{X})$ or $\mathbb{P}_{\mathbf{Y}|\mathbf{X}}$ is another unknown data distribution describing how ground-truth sequences are generated from $\mathbf{X}$. We use contextual encoder h_{Φ} to extract information from $\mathbf{X}$, characterizing by another conditional distribution $\mathbb{P}^{\Phi}(\mathbf{H}|\mathbf{X})$ or $\mathbb{P}_{\mathbf{H}|\mathbf{X}}^{\Phi}$. The predictor g_{Θ} yields a conditional probability $\mathbb{P}^{\Theta}(\mathbf{Y}|\mathbf{H})$ or $\mathbb{P}_{\mathbf{Y}|\mathbf{H}}^{\Theta}$, describing the likelihood estimation of groundtruth $\mathbf{Y}$ under memory $\mathbf{H}$ and parameter Θ .

A Tractable Objective. We begin with restating the objective of the SSIB framework in Sec. 5:

$$\min_{\Theta, \Phi} -\mathbb{I}(\mathbf{Y}; \mathbf{H}) + \beta \mathbb{I}(\mathbf{H}; \mathbf{X}), \quad \text{s.t. } \mathbf{H} \sim \mathbb{P}^{\Phi}(\mathbf{H}|\mathbf{X}). \quad (17)$$

The mutual information terms in Eq. 17 is not computationally tractable. Hence, we seek to obtain an upper bound for it following the approaches in Alemi et al. (2016); Poole et al. (2019).

First, we aim to lower bound $\mathbb{I}(\mathbf{Y}; \mathbf{H})$. Note that $\mathbb{P}(\mathbf{Y}|\mathbf{H})$ is not tractable, so we introduce a variational approximation $\mathbb{P}^{\Theta}(\mathbf{Y}|\mathbf{H})$ for it, leading to:

$$\mathbb{I}(\mathbf{H}; \mathbf{Y}) = \mathbb{E}_{\mathbf{H}, \mathbf{Y}} \left[\log \frac{\mathbb{P}^{\Theta}(\mathbf{Y}|\mathbf{H})}{\mathbb{P}(\mathbf{Y})} \right] + \mathbb{E}_{\mathbf{H}} [D_{\text{KL}}(\mathbb{P}(\mathbf{Y}|\mathbf{H}) \parallel \mathbb{P}^{\Theta}(\mathbf{Y}|\mathbf{H}))] \quad (18)$$

$$\geq \mathbb{E}_{\mathbf{H}, \mathbf{Y}} \left[\log \frac{\mathbb{P}^{\Theta}(\mathbf{Y}|\mathbf{H})}{\mathbb{P}(\mathbf{Y})} \right] \quad (19)$$

$$= \mathbb{E}_{\mathbf{H}, \mathbf{Y}} [\log \mathbb{P}^{\Theta}(\mathbf{Y}|\mathbf{H})] + \mathbb{H}(\mathbf{Y}). \quad (20)$$

The first inequality is according to the non-negativity of KL divergence. Thus, $\mathbb{I}(\mathbf{H}; \mathbf{Y})$ is lower bounded by the expected log-likelihood plus the entropy of $\mathbf{Y}$. The log-likelihood term of Eq. 20

corresponds to the prediction step in SSMs (g_Θ and Eq. 1). The entropy term is independent of Θ and will be ignored in our final objective. Moreover, note that the predicted variable is independent of other variables given $\mathbf{H}_t$: $\mathbb{P}^\Theta(\mathbf{Y}|\mathbf{H}) = \prod_t \mathbb{P}^\Theta(\mathbf{Y}_t|\mathbf{H}_t)$. Therefore, we are able to further expand Eq. 20 as:

$$\mathbb{I}(\mathbf{H}; \mathbf{Y}) \geq \mathbb{E}_{\mathbf{H}, \mathbf{Y}} \left[\sum_{t=1}^T \log \mathbb{P}^\Theta(\mathbf{y}_t|\mathbf{H}_t) \right] + \mathbb{H}(\mathbf{Y}). \quad (21)$$

Second, we upper bound for $\mathbb{I}(\mathbf{H}; \mathbf{X})$. By definition, the mutual information can be written as:

$$\mathbb{I}(\mathbf{H}; \mathbf{X}) = \mathbb{E}_{\mathbf{H}, \mathbf{X}} \left[\log \frac{\mathbb{P}^\Phi(\mathbf{H}|\mathbf{X})}{\mathbb{P}^\Phi(\mathbf{H})} \right], \quad (22)$$

where $\mathbb{P}(\mathbf{H}) = \mathbb{E}_{\mathbf{X}} [\mathbb{P}^\Phi(\mathbf{H}|\mathbf{X})]$ is again intractable. We borrow the technique of Kingma (2013) and introduce a variational approximation $\mathbb{Q}(\mathbf{H})$, resulting in:

$$\mathbb{I}(\mathbf{H}; \mathbf{X}) = \mathbb{E}_{\mathbf{H}, \mathbf{X}} \left[\log \frac{\mathbb{P}^\Phi(\mathbf{H}|\mathbf{X})}{\mathbb{Q}(\mathbf{H})} \right] - D_{\text{KL}}(\mathbb{P}(\mathbf{H})\|\mathbb{Q}(\mathbf{H})) \quad (23)$$

$$\leq \mathbb{E}_{\mathbf{X}} [D_{\text{KL}}(\mathbb{P}^\Phi(\mathbf{H}|\mathbf{X})\|\mathbb{Q}(\mathbf{H}))], \quad (24)$$

which implies $\mathbb{I}(\mathbf{H}; \mathbf{X})$ is upper bounded by the expected KL divergence between the variational approximation and the marginalized distribution. The inequality becomes tighter if $\mathbb{Q}(\mathbf{H})$ approximates $\mathbb{P}^\Phi(\mathbf{H})$ closely. $\mathbb{P}^\Phi(\mathbf{H}|\mathbf{X})$ in the RHS of Eq. 24 functions as the encoding step in SSMs (h_Φ and Eq. 1).

Combining both Eq. 24 and 20, we can recover Eq. 5:

$$\min_{\Theta, \Phi} - \mathbb{E}_{\mathbf{H}, \mathbf{Y}} \left[\sum_{t=1}^T \log \mathbb{P}^\Theta(\mathbf{y}_t|\mathbf{H}_t) \right] + \beta \mathbb{E}_{\mathbf{X}} [D_{\text{KL}}(\mathbb{P}^\Phi(\mathbf{H}|\mathbf{X})\|\mathbb{Q}(\mathbf{H}))], \quad (25)$$

Derivation of KL Divergence Constraints. In this section, we simplify the KL divergence under the setting of Baleen-B. The main idea is to utilize the Markov property and factorize $\mathbf{H}$ along the sequence.

Last, let us fix $\mathbf{X}$ and simplify the KL divergence term:

$$D_{\text{KL}}(\mathbb{P}^\Phi(\mathbf{H}|\mathbf{X})\|\mathbb{Q}(\mathbf{H})) = \mathbb{E}_{\mathbf{H}} \left[\log \frac{\mathbb{P}^\Phi(\mathbf{H}|\mathbf{X})}{\mathbb{Q}(\mathbf{H})} \right] \quad (26)$$

$$= \mathbb{E}_{\mathbf{H}} \left[\log \frac{\mathbb{P}^\Phi(\mathbf{H}_0) \prod_{t=1}^T \mathbb{P}^\Phi(\mathbf{H}_t|\mathbf{H}_{t-1}, \mathbf{X}_{\leq t})}{\mathbb{Q}(\mathbf{H}_0) \prod_{t=1}^T \mathbb{Q}(\mathbf{H}_t|\mathbf{H}_{t-1})} \right] \quad (27)$$

$$= \mathbb{E}_{\mathbf{H}} \left[\log \frac{\mathbb{P}^\Phi(\mathbf{H}_0)}{\mathbb{Q}(\mathbf{H}_0)} + \sum_{t=1}^T \log \frac{\mathbb{P}^\Phi(\mathbf{H}_t|\mathbf{H}_{t-1}, \mathbf{X}_{\leq t})}{\mathbb{Q}(\mathbf{H}_t|\mathbf{H}_{t-1})} \right] \quad (28)$$

$$= \mathbb{E}_{\mathbf{H}_0} \left[\log \frac{\mathbb{P}^\Phi(\mathbf{H}_0)}{\mathbb{Q}(\mathbf{H}_0)} \right] + \sum_{t=1}^T \mathbb{E}_{\mathbf{H}} \left[\log \frac{\mathbb{P}^\Phi(\mathbf{H}_t|\mathbf{H}_{t-1}, \mathbf{X}_{\leq t})}{\mathbb{Q}(\mathbf{H}_t|\mathbf{H}_{t-1})} \right] \quad (29)$$

$$= \sum_{t=1}^T \mathbb{E}_{\mathbf{H}} \left[\log \frac{\mathbb{P}^\Phi(\mathbf{H}_t|\mathbf{H}_{t-1}, \mathbf{X}_{\leq t})}{\mathbb{Q}(\mathbf{H}_t|\mathbf{H}_{t-1})} \right] + \text{const.}, \quad (30)$$

where the last equality holds because $\mathbb{P}^\Phi(\mathbf{H}_0)$ is independent of model parameters.

A.3 SIMPLIFIED KL LOSS IN BALEEN

Baleen-E Given prior $\mathbb{Q}(A_t)$ and $\mathbb{P}(A_t)$, and equation $H_t = H_{t-1} \exp(\mathring{A}_t \text{diag}(\Delta_t)) + B_t x_t \Delta_t$. We can introduce the conditional distribution $\Delta_t|H_t$ and use the KL divergence chain rule:

$$D_{\text{KL}}(\mathbb{P}(\Delta_t) \parallel \mathbb{Q}(\Delta_t)) = \mathbb{E}_{\Delta_t} \left[\log \frac{\mathbb{P}(\Delta_t | H_t) \mathbb{P}(H_t | H_{t-1})}{\mathbb{Q}(\Delta_t | H_t) \mathbb{Q}(H_t)} \right] \quad (31)$$

$$= \mathbb{E}_{\Delta_t | H_t} \mathbb{E}_{H_t} \left[\log \frac{\mathbb{P}(H_t | H_{t-1})}{\mathbb{Q}(H_t)} \right] + \mathbb{E}_{H_t} \mathbb{E}_{\Delta_t | H_t} \left[\log \frac{\mathbb{P}(\Delta_t | H_t)}{\mathbb{Q}(\Delta_t | H_t)} \right] \quad (32)$$

$$= D_{\text{KL}}(\mathbb{P}(H_t | H_{t-1}) \parallel \mathbb{Q}(H_t)) + D_{\text{KL}}(\mathbb{P}(\Delta_t | H_t) \parallel \mathbb{Q}(\Delta_t | H_t)) \quad (33)$$

$$\geq D_{\text{KL}}(\mathbb{P}(H_t | H_{t-1}) \parallel \mathbb{Q}(H_t)) \quad (34)$$

Baleen-B Given prior $\mathbb{Q}(A_t)$ and $\mathbb{P}(A_t)$, and equation $H_t = H_{t-1}A_t + B_tx_t$, we obtain the following relation, analogous to the derivation in Baleen-B:

$$D_{\text{KL}}(\mathbb{P}(A) \parallel \mathbb{Q}(A_t)) = D_{\text{KL}}(\mathbb{P}(H_t | H_{t-1}) \parallel \mathbb{Q}(H_t)) + D_{\text{KL}}(\mathbb{P}(A_t | H_t) \parallel \mathbb{Q}(A_t | H_t)) \quad (35)$$

$$= D_{\text{KL}}(\mathbb{P}(H_t | H_{t-1}) \parallel \mathbb{Q}(H_t)) \quad (36)$$

In this case, the inequality becomes equality because $D_{\text{KL}}(\mathbb{P}(A_t | H_t) \parallel \mathbb{Q}(A_t | H_t)) = 0$ which follows directly from the one-to-one mapping between A_t and H_t given H_{t-1} .

B EXPERIMENT DETAILS

B.1 ADVERSARIAL ROBUSTNESS.

Position Bias Attack. In this experiment, we train all the compared models using the framework from (Arora et al., 2023). Each model consists of three layers, with a hidden state dimension of 32 per layer. Training is conducted for 100 epochs on the CIFAR-10 dataset. For our Baleen-B model, we set $\beta = 0.01$, while for the Baleen-E model, $\beta = 0.5$.

Prompt Attack. The following text prompt is used to query the language models in this experiment, where the placeholder ‘{content}’ will be replaced with two sentences. The second sentence provides an explanation of the pronoun used in the first sentence. The language model is then tasked with predicting whether the pronoun resolution is “correct” or “incorrect.”

In the following sentence, does the hypothesis correctly resolve the ambiguous pronoun? Answer ‘correct’ or ‘incorrect’. Please classify:
Question: {content}
Answer:

Below is a negative sample of the two sentences:

Sentence 1: The drain is clogged with hair. It has to be cleaned.
Sentence 2: The hair has to be cleaned.