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'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 points at 370M and 1.38 points at 7B. Baleen is also more robust to localized perturbations and adversarial attacks: under sequence perturbations, it reduces the accuracy drop by over 15%. Finally, Baleen's self-interpretations achieve higher average fidelity than IG and Grad-CAM across text classification tasks. Overall, despite the robustness–accuracy and interpretability–accuracy trade-offs, Baleen shifts the Pareto frontier outward, delivering better robustness, interpretability, and accuracy.

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 $\boldsymbol{H}_{t-1}$, which maintains a running summarization of past token embeddings $\boldsymbol{X}_{1:t-1}$ for prediction, and *ii)* a parameterized recurrence, which applies a *state transition* matrix $\boldsymbol{A}_t$ 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 $\boldsymbol{A}_t$ involves the input token embedding $\boldsymbol{X}_t$.

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 increasing 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).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

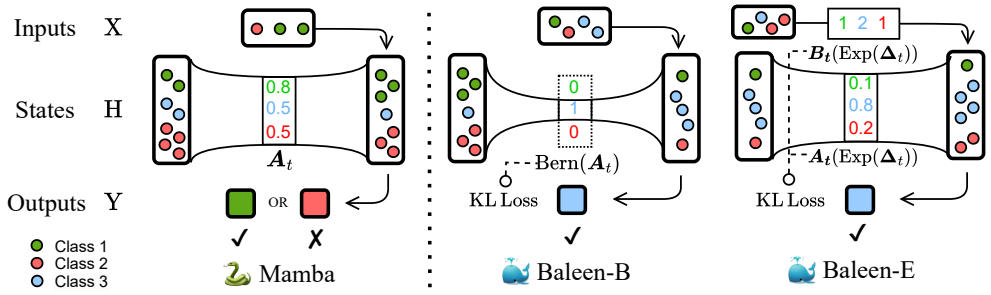


Figure 1: Overview of Baleen compared with previous SSMs such as Mamba. The information bottleneck injects random noise and enhances the selection of the state transition matrix by adapting the deterministic $\boldsymbol{A}_t$ or $\boldsymbol{\Delta}_t$ to the randomly sampled variable. For simplicity, we omit deterministic $\boldsymbol{B}_t$ and $\boldsymbol{C}_t$.

To address these limitations, we introduce *Baleen*, a family of information-bottleneck SSM architectures that explicitly optimizes for the context compression rate through a 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 $\mathbb{I}(\boldsymbol{Y};\boldsymbol{H})$ without penalizing $\mathbb{I}(\boldsymbol{H};\boldsymbol{X})$, 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 $\boldsymbol{X} := [\boldsymbol{x}_1^\top, \cdots, \boldsymbol{x}_T^\top]^\top \in \mathbb{R}^{T\times D}$ 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 $\{(\boldsymbol{A}_t, \boldsymbol{B}_t, \boldsymbol{C}_t, \boldsymbol{\Delta}_t)\}_{t\in[T]}$ to process the sequence according to the following equations:

$$\boldsymbol{H}_t = \boldsymbol{A}_t \odot \boldsymbol{H}_{t-1} + \boldsymbol{B}_t \odot (\boldsymbol{1}_N \boldsymbol{x}_t^\top), \quad \widehat{\boldsymbol{y}}_t = \boldsymbol{C}_t^\top \boldsymbol{H}_t \tag{1}$$

where $\odot$ denotes element-wise multiplication, $\boldsymbol{A}_t = \exp(\mathring{\boldsymbol{A}}_t \operatorname{diag}(\boldsymbol{\Delta}_t))$, $\boldsymbol{B}_t = \mathring{\boldsymbol{B}}_t \boldsymbol{\Delta}_t^\top$, $\boldsymbol{\Delta}_t \in \mathbb{R}_+^D$, $\mathring{\boldsymbol{A}}_t \in \mathbb{R}_-^{N\times D}$ $\mathring{\boldsymbol{B}}_t \in \mathbb{R}^N$, $\boldsymbol{C}_t \in \mathbb{R}^{N\times D}$ for all $t \in [T]$. Note that $\boldsymbol{\Delta}_t$ is strictly positive plus $\mathring{\boldsymbol{A}}_t$ is

strictly negative, ensuring $\boldsymbol{A}_t \in (0,1)^{N\times D}$. Define $\boldsymbol{H} := [\boldsymbol{H}_t \in \mathbb{R}^{N\times D}]_{t\in[T]}$ as a $T \times N \times D$ tensor to represent the intermediate *memory states*, and $\widehat{\boldsymbol{Y}} := [\widehat{\boldsymbol{y}}_1^\top, \cdots, \widehat{\boldsymbol{y}}_T^\top]^\top \in \mathbb{R}^{T\times D}$ denotes the output sequence. Next, the operation $\boldsymbol{B}_t \odot (\boldsymbol{1}_N \boldsymbol{x}_t^\top)$ encodes the input tokens into the hidden state space. And finally, $\boldsymbol{C}_t^\top \boldsymbol{H}_t$ decodes the memory state to generate the prediction $\widehat{\boldsymbol{y}}_t$ for the t-th token.

In *S4* (Gu et al., 2021a), the parameters $\{(\mathring{\boldsymbol{A}}_t, \mathring{\boldsymbol{B}}_t, \boldsymbol{C}_t, \boldsymbol{\Delta}_t)\}_{t\in[T]}$ 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:

$$\mathring{\boldsymbol{A}}_t = \boldsymbol{A}, \quad \mathring{\boldsymbol{B}}_t = \boldsymbol{W}_B \boldsymbol{x}_t, \quad \boldsymbol{C}_t = (\boldsymbol{W}_C \boldsymbol{x}_t)\boldsymbol{1}_D^\top, \quad \boldsymbol{\Delta}_t = \mathrm{softplus}(\boldsymbol{W}_\Delta \boldsymbol{x}_t) \tag{2}$$

where the learnable parameters in an SSM layer includes $\boldsymbol{A} \in \mathbb{R}_-^{N\times D}$ being invariant across tokens, $\boldsymbol{W}_B \in \mathbb{R}^{N\times D}$, $\boldsymbol{W}_C \in \mathbb{R}^{N\times D}$, and $\boldsymbol{W}_\Delta \in \mathbb{R}^{D\times D}$. In Mamba2, Dao & Gu (2024) proposes to impose additional structures on the parameter space. Specifically, $\boldsymbol{A}$ is simplified to $\boldsymbol{1}_N \boldsymbol{\alpha}^\top$ for some $\boldsymbol{\alpha} \in \mathbb{R}^D$ and $\boldsymbol{W}_\Delta$ 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:

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

where β is a hyperparameter to control the balance, $\boldsymbol{Z}$ follows conditional distribution $\mathbb{P}^\Phi(\boldsymbol{Z}|\boldsymbol{X})$ parametrized by an encoder Φ and $\boldsymbol{Y}$ follows conditional distribution $\mathbb{P}^\Theta(\boldsymbol{Y}|\boldsymbol{H})$ 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 dereived a tractable variation upper bound (up to a constant) as below:

$$\min_{\Theta,\Phi} -\mathbb{E}\left[\log \mathbb{P}^\Theta(\boldsymbol{Y}|\boldsymbol{Z})\right] + \beta\, \mathbb{E}\left[\mathrm{KL}(\mathbb{P}^\Phi(\boldsymbol{Z}|\boldsymbol{X}) \| \mathbb{Q}(\boldsymbol{Z}))\right] \tag{4}$$

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 incorporate principled randomness modeling within the SSIB framework. Finally, in Sec. 3.3, we demonstrate how our architectures are 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 $\mathbb{P}(\boldsymbol{X},\boldsymbol{Y}) = \mathbb{P}(\boldsymbol{Y}|\boldsymbol{X})\mathbb{P}(\boldsymbol{X})$[1], which is supported on $\mathcal{X}$ and $\mathcal{Y}$. In the case of causal language modeling (Radford et al., 2018; 2019; Brown et al., 2020), $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}) = \delta_{\boldsymbol{X}^*}(\boldsymbol{Y})$, where $\boldsymbol{X}^* = [\boldsymbol{x}_2^\top, \cdots, \boldsymbol{x}_T^\top, \mathsf{EOS}]^\top$ is a shifted version of $\boldsymbol{X}$ with EOS (special token) padding.

[1]When 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.

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

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:

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

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)*. Beyond compression, the trajectory-level KL term in Eq. 5 controls a high-probability upper bound on the generalization gap of the induced stochastic predictor (Proposition B.1), providing a principled justification for SSIB beyond likelihood fitting.

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): $\max_{\Theta,\Phi} \mathbb{E}_{\boldsymbol{X},\boldsymbol{Y}}[\sum_{t=1}^{T-1} \log \mathbb{P}^{\Theta,\Phi}(\boldsymbol{y}_t|\boldsymbol{X}_{\leq t})]$. We argue that simply maximizing the mutual information between $\boldsymbol{y}_t$ and $\boldsymbol{H}_t$ only ensures $\boldsymbol{H}_t$ retains sufficient information for predicting $\boldsymbol{y}_t$. However, this does not guarantee that $\boldsymbol{H}_t$ achieves the maximal compression of $\boldsymbol{X}_{\leq t}$ and only focuses on tokens useful for prediction, potentially leading to information loss as the context length grows larger. Moreover, $\boldsymbol{H}_t$ may capture *spurious correlations* between $\boldsymbol{X}_{\leq t}$ and $\boldsymbol{y}_t$, making the predictions highly susceptible to noise (Chen et al., 2018; Wang et al., 2024). Instead, our approach explicitly models the stochastic relationship between $\boldsymbol{X}$ and $\boldsymbol{H}$, enabling the denoising of spurious patterns in $\boldsymbol{H}$—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., $\boldsymbol{A}_t$) is the key component to randomize, whether randomness should also be incorporated into the new token encoding (i.e., $\boldsymbol{B}_t$) 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 $\boldsymbol{\Delta}_t$. The overall architectural design of SSIB is illustrated in Fig. 1.

Bernoulli-Distributed Transition. As mentioned before, each element in the memory state $\boldsymbol{H}_t$ 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 natural to model $\boldsymbol{A}_t$ as a variable follows Bernoulli distribution. In this case, $\boldsymbol{H}_t$ is an affine transformation of $\boldsymbol{A}_t$ and also follows a Bernoulli distribution. Moreover, each outcome of $\boldsymbol{H}_t$ corresponds to a unique outcome of $\boldsymbol{A}_t$ with the same probability. Therefore, the KL divergence is preserved under this affine transformation, i.e., the following equation holds:

$$\mathrm{KL}(\mathbb{P}(\boldsymbol{H}_t|\boldsymbol{H}_{t-1}, \boldsymbol{X}_{\leq t}) \| \mathbb{Q}(\boldsymbol{H}|\boldsymbol{H}_{t-1})) = \mathrm{KL}(\mathbb{P}(\boldsymbol{A}_t) || \mathbb{Q}(\boldsymbol{A}_t)) \tag{6}$$

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

[2]Without 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.

Because variational bound Eq. 5 holds true for any prior distribution $\mathbb{Q}$, we can simply define the prior distribution of $\boldsymbol{A}_t$ as $\mathrm{Bern}(p_t \mathbf{1}_N \mathbf{1}_D^\top)$ 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:

$$L_{\mathrm{KL}} = \mathbb{E}\Big[\sum_{t,n,d} \boldsymbol{A}_t^{(n,d)} \log \frac{\boldsymbol{A}_t^{(n,d)}}{p_t} + (1 - \boldsymbol{A}_t^{(n,d)}) \log \frac{1 - \boldsymbol{A}_t^{(n,d)}}{1 - p_t}\Big] \tag{7}$$

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 $\boldsymbol{A}_t$ and $\boldsymbol{B}_t$ is challenging, especially $\boldsymbol{A}$ and $\boldsymbol{B}_t$ should conform to different distributions even discrete and continuous are not the same. Based on our observation that both $\boldsymbol{A}$ and $\boldsymbol{B}_t$ are parameterized with $\boldsymbol{\Delta}_t$, we may assume $\boldsymbol{\Delta}_t$ to be a random variable to implicitly affect both terms. In SSMs, $\boldsymbol{\Delta}_t$ 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 $\boldsymbol{\Delta}_t$ in Eq. 2 as a random variable following Exponential distribution. In this case, $\boldsymbol{H}_t$ has an explicit expression form:

$$\boldsymbol{H}_t = \boldsymbol{H}_{t-1} \exp(\mathring{\boldsymbol{A}}_t \operatorname{diag}(\boldsymbol{\Delta}_t)) + \mathbf{1}_N \boldsymbol{x}_t \mathring{\boldsymbol{B}}_t \boldsymbol{\Delta}_t \tag{8}$$

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 $\boldsymbol{H}_t$ as a function of $\boldsymbol{\Delta}_t$ is not invertible, the KL divergence in Eq. 5 cannot be directly reduced to that between $\boldsymbol{\Delta}_t$ 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 $\boldsymbol{\Delta}_t$. Further details are provided in Appendix A.3.

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

$$L_{\mathrm{KL}} = \mathbb{E}\left[\sum_{t,d} \log \frac{\boldsymbol{\Delta}_t^{(d)}}{\lambda_t} - (\boldsymbol{\Delta}_t^{(d)} - \lambda_t) \frac{1}{\boldsymbol{\Delta}_t^{(d)}}\right], \tag{9}$$

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 $\boldsymbol{\Delta}_t$ tends to infinity, the first term in Eq. 8 vanishes due to the negative $\mathring{\boldsymbol{A}}_t$, and $\boldsymbol{H}_t$ depends solely on the new token embeddings. Conversely, when $\boldsymbol{\Delta}_t$ tends to zero the second term vanishes, and $\boldsymbol{H}_t$ is entirely inherited from past memory. Therefore, $\boldsymbol{\Delta}_t$ 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. Empirically, this localized stochastic gating yields concrete compression: IB-designated layers exhibit roughly $2\times$ higher gate sparsity and lower effective rank than the corresponding Mamba2 layers, while non-IB layers remain unchanged (Appendix D.3).

3.3 Inherent Interpretability

As previously discussed, each element in the state transition matrix $\boldsymbol{A}_t$ 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.

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

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

Under a monotone linear-readout approximation, S_t is an order-preserving proxy for the history-retention contribution of token t to future logits (Proposition B.4): a larger S_t implies that more of the pre-t memory is preserved and allowed to influence downstream predictions, giving a mechanistic justification for treating S_t as a token-level importance score. To operationalize interpretation during inference, we first compute the expectation of the stochastic transition matrices $\boldsymbol{A}_t$ 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 evaluate our approach on standard language benchmarks to assess (i) accuracy, (ii) the faithfulness of its inherent interpretations, and (iii) robustness to input perturbations. Notably, Baleen improves trustworthiness without an accuracy drop, contrary to the commonly observed robustness–accuracy (Zhang et al., 2019) and interpretability–accuracy (Rudin, 2019) trade-offs.

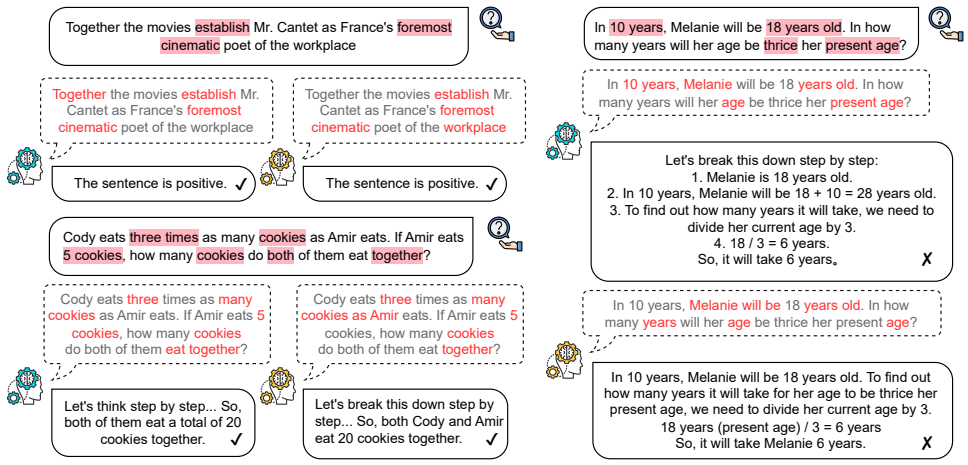
Settings. We pre-trained 370M Baleen-B and Baleen-E models on top of the Mamba2 architecture (Dao & Gu, 2024) using 20B tokens from SlimPajama (Soboleva et al., 2023), and fine-tuned their 7B counterparts from the pre-trained Mamba2-Codestral checkpoint (AI@Mistral, 2024) on 0.5B instruction-following tokens from Crystal (Liu et al., 2023); for fair comparison, the Mamba2 7B baseline underwent the same continual fine-tuning. The KL weight β is set to 10^{-1} for Baleen-B and 10^{-6} for Baleen-E, selected via the ablation in Appendix D.2. We compare primarily against vanilla Mamba2, and additionally include several state space and linear attention baselines: RetNet (Sun et al., 2023), RWKV6 (Peng et al., 2023), GLA (Yang et al., 2023), Mamba (Gu & Dao, 2023), and the recently proposed Gated DeltaNet (Yang et al., 2024a).[3]

4.1 Language Benchmarks

Pre-trained 370M Models. Baleen-B and Baleen-E achieve the best overall performance, with average accuracies of **33.75%** and **33.86%**, surpassing all baselines including the strong Gated-DeltaNet (33.23%)—the architecture behind Qwen3-Next (Qwen-Team, 2025). The gains concentrate on commonsense reasoning, where Baleen-E leads ARC-Easy (**39.35%**, +1.81 over Mamba2) and PIQA (**60.45%**, +0.49) and Baleen-B improves on Mamba2 on Winogrande (**51.22%**, +0.55), with marginal improvements on language understanding (MMLU 22.97% / 23.01% vs. 22.92%). These results confirm the benefit of our design, and the advantage extends to hybrid settings: interleaving Baleen-B with Multi-head Latent Attention (MLA) further surpasses hybrid Gated DeltaNet (Appendix D.1).

Fine-tuned 7B Models. After instruction tuning, Baleen-B attains the highest average of **43.24%**, followed by Baleen-E (**42.43%**), both ahead of Mamba2 (41.86%). Baleen-B excels on knowledge-

[3]We compare only to Mamba2 at 7B, as no checkpoints of comparable quality to Mamba2-Codestral for the other architectures are publicly released on HuggingFace.



(a) Right answer interpretation. All important tokens are captured. (b) Wrong answer interpretation. 18 and thrice are not captured.

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.

intensive tasks (BoolQ **73.06%**, +10.43; MMLU **34.76%**, +1.92), while Baleen-E leads on commonsense (PIQA **68.28%**, +1.65; HellaSwag **40.26%**, +2.2). Both also surpass Mamba2 on Winogrande (58.01% / 56.99% vs. 57.14%). Overall, both Baleen variants consistently outperform the Mamba2 baseline, with stronger gains emerging at scale.

4.2 Inherent Interpretation

Quantitative Evaluation. We quantitatively evaluate model interpretations and report the results in Tab. 2. Fidelity AUC (Zhu et al., 2024; Jafari et al., 2024) is commonly used to reduce the evaluation bias caused by choosing a single sparsity value, e.g., Fidelity@k, which may lead to inconsistent comparisons. We evaluate on four well-established text classification tasks, including SNLI (Bowman et al., 2015), IMDb (Maas et al., 2011), SST2 (Socher et al., 2013), and Rotten Tomatoes (Pang & Lee, 2005), using fine-tuned pretrained 370M models. We compare our self-interpretation method with widely 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 a fair comparison, we keep the interpreted model fixed and apply these post-hoc methods to the corresponding Baleen-B and Baleen-E 370M models. We further include a vanilla Mamba2 model as a cross-architecture reference.

As shown in Tab. 2, both Baleen-B and Baleen-E produce more faithful explanations than post-hoc baselines on the same architectures. Their self-interpretation scores also outperform the strongest post-hoc result on vanilla Mamba2, suggesting that the improvement does not come merely from architectural differences. Notably, our method is intrinsically self-interpretable and produces explanations without additional post-hoc attribution cost, enabling efficient monitoring of model behavior during inference.

Table 2: Explanation fidelity on fine-tuned 370M models. "RT." refers to Rotten Tomatoes. "Avg." denotes the macro average over four datasets. Mamba2 is evaluated with post-hoc methods only.

Method	**Mamba2** (post-hoc only)					**Baleen-B**					**Baleen-E**				
	SNLI	IMDb	SST2	RT.	Avg.	SNLI	IMDb	SST2	RT.	Avg.	SNLI	IMDb	SST2	RT.	Avg.
IG	43.56	**57.23**	**50.37**	38.18	**47.34**	44.00	40.21	39.94	40.50	41.16	24.23	2.45	38.47	2.27	16.86
GradX	**50.58**	36.06	43.99	**42.19**	43.21	48.85	42.98	47.96	40.94	45.18	48.88	43.69	47.79	41.63	45.50
GradCAM	49.33	44.49	38.19	37.68	42.42	54.98	**49.59**	42.57	42.00	47.29	**54.73**	**52.69**	39.19	48.65	48.82
Post-hoc Avg.					44.32					44.54					37.06
Self (Ours)			–			**55.81**	45.24	**57.65**	**51.49**	**52.55**	54.31	51.66	**62.50**	**62.17**	**57.66**

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

Models	**Perturbation region** (seq. length = 1024) [0:0]	[0:32]	[0:96]	[928:1024]	[992:1024]	Avg.
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%)	0.332 (↓ 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	0.659	0.634 (↓ 3.82%)	0.603 (↓ 8.58%)	0.314 (↓ 52.37%)	0.453 (↓ 31.23%)	0.532±0.146
Baleen-E	0.643	0.639 (↓ **0.57%**)	0.628 (↓ 2.33%)	**0.365** (↓ **43.16%**)	0.433 (↓ 32.63%)	**0.542±0.132**

Qualitative Example. We probe the fine-tuned 7B models on SST and GSM8K and visualize three samples in Fig. 2, with human-annotated salient tokens from Wu et al. (2023) as reference. As Fig. 2(a) shows, both Baleen-E (blue) and Baleen-B (yellow) recover the human-annotated rationales when the answer is correct: $3/4$ positive-sentiment tokens for SST, and the key objects, numbers, and the cue word "together" for the arithmetic problem. Conversely, Fig. 2(b) traces a wrong answer back to inattention on "18" and "thrice"—the model treats Melanie's current age as 18 rather than 8 and divides instead of multiplying—illustrating how Baleen's interpretations enable early error diagnosis during long-form generation.

4.3 Adversarial Robustness

Before reporting empirical results, we briefly note the mechanism underlying Baleen's robustness gains. Under mild Lipschitz assumptions, an input perturbation δ_s propagates to the hidden state at time t damped by the multiplicative factor $\prod_{j=s+1}^{t} \rho_j$, where ρ_j is the effective retention coefficient at step j (Proposition B.2). SSIB regularization contracts ρ_j—directly via the Bernoulli gates in Baleen-B, and via larger Δ_t shrinking $A_t = \exp(\mathring{A}_t \mathrm{diag}(\Delta_t))$ in Baleen-E—thereby suppressing the long-range accumulation of perturbations through recurrent memory.

Position Bias Attack. SSMs exhibit position bias, which can be exploited to compromise performance (Wang et al., 2024). We probe this on *image classification over pixel sequences* (Tay et al., 2020): each $W \times H$ image is flattened into WH RGB tokens, with a learnable class token appended at the end as in ViT (Dosovitskiy et al., 2021). Since predictions read from this trailing class token, models with weaker robustness are particularly vulnerable to perturbations in trailing tokens; we therefore corrupt both leading and trailing spans on CIFAR-10.

As shown in Tab. 3, both Baleen variants exhibit balanced sensitivity across positions. Under leading-token corruption (`[0:32]`), Baleen-B and Baleen-E retain accuracy comparable to vanilla Mamba while clearly outperforming Mamba2. More strikingly, trailing-token corruption (`[992:1024]`) collapses Mamba and Mamba2 by over 70%, whereas SSIB limits the drop to $\sim 30\%$. Baleen-E attains the highest average accuracy overall, confirming that stochastic gating mitigates position-bias-induced fragility.

Prompt Attack. To further evaluate robustness, we conduct prompt-attack experiments on our fine-tuned 7B models using WNLI and SST2 (Wang et al., 2018). Following PromptBench (Zhu et al., 2023a), we apply four perturbations spanning character- to sentence-level attacks: DeepWordBug, TextBugger, CheckList, and StressTest (Zhu et al., 2023b), and report the performance drop ratio (PDR) between normal and perturbed prompts.

As shown in Tab. 4, both Baleen-E and Baleen-B consistently achieve lower drop ratios than Mamba2 across most adversarial settings, and the two variants complement each other: Baleen-B is most resilient on SST2 under Checklist and StressTest, while Baleen-E dominates on WNLI under Checklist and TextBugger (drop ratios as low as 17.65 and 0.00). This indicates that SSIB regularization improves robustness across attack types ranging from character to sentence level.

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

5 Related Work

Linear State Space Modeling as Online Learning. Recent works (Sun et al., 2024; Liu et al., 2024; Behrouz et al., 2024) unify linear SSMs from an online-learning perspective, where the recurrence is viewed as one step of online SGD on a state-update objective; this subsumes recurrence designs including gating (Sun et al., 2023; Gu & Dao, 2023; Yang et al., 2023; Peng et al., 2023; De et al., 2024; Goldstein et al., 2024), the delta rule (Schlag et al., 2021; Yang et al., 2024b;a), and momentum (Behrouz et al., 2024). Our SSIB framework is complementary: rather than prescribing the update dynamics, it regularizes $\mathbf{H}_t$ against a prior $\mathbb{Q}$ via an information-bottleneck objective. As Linear Attention (Qin et al., 2022; Sun et al., 2023) is a special case of linear SSMs, Baleen extends to broader linear-recurrent architectures (MiniMax et al., 2025; Ren et al., 2023; 2024; Lieber et al., 2024; Ma et al., 2022; 2024).

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) 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 treated 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., 2020), 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 that Baleen achieves a new Pareto frontier in accuracy, interpretability, and robustness. We note two limitations of this they characterize SSIB's effect under simplifying assumptions, work: our theoretical analyses (Appendix. B) are partial: rather than establishing certified guarantees over the full nonlinear predictor; and our empirical validation is confined to the Mamba/Mamba2 family at up to 7B parameters, leaving extension to other linear-recurrent backbones and to frontier scale as future work.

AI USE STATEMENT

In this work, we used generative AI tools for limited discussion and sanity-checking of theoretical estimates. We did not use generative AI tools to formulate research hypotheses, design the methodology or experiments, implement the proposed methods, analyze or interpret experimental results, or generate synthetic data; the remaining required-disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to improve the readability and presentation of the manuscript.

We reviewed all AI-assisted work. In particular, all mathematical derivations, theoretical claims, and estimates were independently derived and verified by the authors, and all AI-assisted edits to the manuscript were reviewed by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

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 Zimerman, 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 Bennetot, 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 Timalsina, 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.

Jamba Team, Barak Lenz, Alan Arazi, et al. Jamba-1.5: Hybrid transformer-mamba models at scale. *arXiv preprint arXiv:2408.12570*, 2024.

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 Khabsa. 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.

Aakash Lahoti, Kevin Li, Berlin Chen, Caitlin Wang, Aviv Bick, J Zico Kolter, Tri Dao, and Albert Gu. Mamba-3: Improved sequence modeling using state space principles. In *The Fourteenth International Conference on Learning Representations*, 2026.

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.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 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, Qiuhui 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&noteId=okbDFg9W0n.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. *15th European Signal Processing Conference*, pp. 606–610, 2007.

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. PMlR, 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*, 2020.

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.

Aolin Xu and Maxim Raginsky. Information-theoretic analysis of generalization capability of learning algorithms. *Advances in neural information processing systems*, 30, 2017.

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.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pp. 7472–7482. PMLR, 2019.

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 Derivations

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:

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

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:

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

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 $\mathbb{Q}(A)$:

$$\mathrm{KL}(\mathbb{P}(A)\|\mathbb{Q}(A)) = \mathbb{E}_{A\sim\mathbb{P}}\left[\log \frac{\mathbb{P}(A)}{\mathbb{Q}(A)}\right]. \tag{13}$$

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:

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

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

$$\mathbb{I}(A;B) = \mathrm{KL}(\mathbb{P}(A,B)\|\mathbb{P}(A)\mathbb{P}(B)). \tag{16}$$

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 $\boldsymbol{X}$, memory states $\boldsymbol{H}$, and ground-truth sequence $\boldsymbol{Y}$. First of all, $\mathbb{P}(\boldsymbol{X})$ or $\mathbb{P}_{\boldsymbol{X}}$ is an unknown distribution where input sequences are sampled from. The conditional distribution $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X})$ or $\mathbb{P}_{\boldsymbol{Y}|\boldsymbol{X}}$ is another unknown data distribution describing how ground-truth sequences are generated from $\boldsymbol{X}$. We use contextual encoder h_Φ to extract information from $\boldsymbol{X}$, characterizing by another conditional distribution $\mathbb{P}^\Phi(\boldsymbol{H}|\boldsymbol{X})$ or $\mathbb{P}^\Phi_{\boldsymbol{H}|\boldsymbol{X}}$. The predictor g_Θ yields a conditional probability $\mathbb{P}^\Theta(\boldsymbol{Y}|\boldsymbol{H})$ or $\mathbb{P}^\Theta_{\boldsymbol{Y}|\boldsymbol{H}}$, describing the likelihood estimation of groundtruth $\boldsymbol{Y}$ under memory $\boldsymbol{H}$ and parameter Θ.

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

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

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

$$\mathbb{I}(\boldsymbol{H};\boldsymbol{Y}) = \mathbb{E}_{\boldsymbol{H},\boldsymbol{Y}}\left[\log\frac{\mathbb{P}^{\Theta}(\boldsymbol{Y}|\boldsymbol{H})}{\mathbb{P}(\boldsymbol{Y})}\right] + \mathbb{E}_{\boldsymbol{H}}\left[\mathrm{KL}(\mathbb{P}(\boldsymbol{Y}|\boldsymbol{H})\|\mathbb{P}^{\Theta}(\boldsymbol{Y}|\boldsymbol{H}))\right] \tag{18}$$

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

$$= \mathbb{E}_{\boldsymbol{H},\boldsymbol{Y}}\left[\log\mathbb{P}^{\Theta}(\boldsymbol{Y}|\boldsymbol{H})\right] + \mathbb{H}(\boldsymbol{Y}). \tag{20}$$

The first inequality is according to the non-negativity of KL divergence. Thus, $\mathbb{I}(\boldsymbol{H};\boldsymbol{Y})$ is lower bounded by the expected log-likelihood plus the entropy of $\boldsymbol{Y}$. 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 $\boldsymbol{H}_t$: $\mathbb{P}^{\Theta}(\boldsymbol{Y}|\boldsymbol{H}) = \prod_t \mathbb{P}^{\Theta}(\boldsymbol{Y}_t|\boldsymbol{H}_t)$. Therefore, we are able to further expand Eq. 20 as:

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

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

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

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

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

$$\leq \mathbb{E}_{\boldsymbol{X}}\left[\mathrm{KL}(\mathbb{P}^{\Phi}(\boldsymbol{H}|\boldsymbol{X})\|\mathbb{Q}(\boldsymbol{H}))\right], \tag{24}$$

which implies $\mathbb{I}(\boldsymbol{H};\boldsymbol{X})$ is upper bounded by the expected KL divergence between the variational approximation and the marginalized distribution. The inequality becomes tighter if $\mathbb{Q}(\boldsymbol{H})$ approximates $\mathbb{P}^{\Phi}(\boldsymbol{H})$ closely. $\mathbb{P}^{\Phi}(\boldsymbol{H}|\boldsymbol{X})$ 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:

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

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 $\boldsymbol{H}$ along the sequence.

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

$$\mathrm{KL}(\mathbb{P}^{\Phi}(\boldsymbol{H}|\boldsymbol{X})\|\mathbb{Q}(\boldsymbol{H})) = \mathbb{E}_{\boldsymbol{H}}\left[\log\frac{\mathbb{P}^{\Phi}(\boldsymbol{H}|\boldsymbol{X})}{\mathbb{Q}(\boldsymbol{H})}\right] \tag{26}$$

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

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

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

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

where the last equality holds because $\mathbb{P}^{\Phi}(\boldsymbol{H}_0)$ 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 $H_t = H_{t-1}\exp(\mathring{A}_t \mathrm{diag}(\Delta_t)) + B_t x_t \Delta_t$. We can introduce the conditional distribution $\Delta_t|H_t$ and use the KL divergence chain rule:

$$\begin{aligned}
\mathrm{KL}(\mathbb{P}(\Delta_t)\|\mathbb{Q}(\Delta_t) &= \mathbb{E}_{\boldsymbol{\Delta_t}}\left[\log \frac{\mathbb{P}(\Delta_t|H_t)\mathbb{P}(H_t|H_{t-1})}{\mathbb{Q}(\Delta_t|H_t)\mathbb{Q}(H_t)}\right] && (31)\\
&= \mathbb{E}_{\boldsymbol{\Delta_t}|\boldsymbol{H_t}}\,\mathbb{E}_{\boldsymbol{H_t}}\left[\log \frac{\mathbb{P}(H_t|H_{t-1})}{\mathbb{Q}(H_t)}\right] + \mathbb{E}_{\boldsymbol{H_t}}\,\mathbb{E}_{\boldsymbol{\Delta_t}|\boldsymbol{H_t}}\left[\log \frac{\mathbb{P}(\Delta_t|H_t)}{\mathbb{Q}(\Delta_t|H_t)}\right] && (32)\\
&= \mathrm{KL}(\mathbb{P}(H_t|H_{t-1})\|\mathbb{Q}(H_t) + \mathrm{KL}(\mathbb{P}(\Delta_t|H_t)\|\mathbb{Q}(\Delta_t|H_t) && (33)\\
&\geq \mathrm{KL}(\mathbb{P}(H_t|H_{t-1})\|\mathbb{Q}(H_t) && (34)
\end{aligned}$$

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

$$\begin{aligned}
\mathrm{KL}(\mathbb{P}(A_t)\|\mathbb{Q}(A_t)) &= \mathrm{KL}(\mathbb{P}(H_t|H_{t-1})\|\mathbb{Q}(H_t)) + \mathrm{KL}(\mathbb{P}(A_t|H_t)\|\mathbb{Q}(A_t|H_t)) && (35)\\
&= \mathrm{KL}(\mathbb{P}(H_t|H_{t-1})\|\mathbb{Q}(H_t)) && (36)
\end{aligned}$$

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

B Theoretical Analysis

B.1 Generalization

We provide a partial information-theoretic explanation for why the SSIB objective can improve generalization. Our analysis is stated for the *induced stochastic predictor*, namely

$$G_{\Theta,\Phi}(X): \qquad H \sim P_\Phi(H \mid X), \qquad \widehat{Y} \sim P_\Theta(\cdot \mid H),$$

rather than for the final deterministic predictor obtained by fixing a single latent trajectory. This allows us to directly leverage the stochastic encoder structure of SSIB.

For a sequence input $X = [x_1^\top, \ldots, x_T^\top]^\top$, define the trajectory-level posterior and prior by

$$P_\Phi(H \mid X) := \prod_{t=1}^{T} P_\Phi(H_t \mid H_{t-1}, X_{\leq t}), \tag{37}$$

$$Q(H) := \prod_{t=1}^{T} Q(H_t \mid H_{t-1}), \tag{38}$$

where H_0 is fixed or sampled from a data-independent initialization distribution. We further define the sequential information complexity induced by SSIB:

$$\mathcal{C}_{\mathrm{SSIB}} := \mathbb{E}_X\left[\sum_{t=1}^{T} \mathrm{KL}(P_\Phi(H_t \mid H_{t-1}, X_{\leq t}) \parallel Q(H_t \mid H_{t-1}))\right]. \tag{39}$$

The following proposition connects $\mathcal{C}_{\mathrm{SSIB}}$ to the generalization gap of the induced stochastic predictor.

Proposition B.1 (Generalization of the induced stochastic predictor)**.** *Assume the per-example loss $\ell(G_{\Theta,\Phi}(X), Y)$ is bounded in $[0,1]$, and let*

$$R(\Theta,\Phi) := \mathbb{E}_{(X,Y)\sim\mathcal{D}}[\ell(G_{\Theta,\Phi}(X), Y)], \qquad \widehat{R}_n(\Theta,\Phi) := \frac{1}{n}\sum_{i=1}^{n} \ell(G_{\Theta,\Phi}(X_i), Y_i),$$

where $S = \{(X_i, Y_i)\}_{i=1}^n$ is an i.i.d. sample from $\mathcal{D}$. Then there exists an absolute constant $c > 0$ such that, with probability at least $1-\delta$ over the draw of S,

$$R(\Theta,\Phi) - \widehat{R}_n(\Theta,\Phi) \;\leq\; c\sqrt{\frac{\mathcal{C}_{\mathrm{SSIB}} + \log(1/\delta)}{n}}. \tag{40}$$

Proof sketch. The predictor $G_{\Theta,\Phi}$ is a Gibbs-type predictor induced by the randomized encoder $P_\Phi(H \mid X)$. Standard randomized-encoder / information-complexity generalization arguments (Xu & Raginsky, 2017) imply that the high-probability generalization gap is controlled by a trajectory-level KL term of the form

$$\mathbb{E}_X \, \mathrm{KL}(P_\Phi(H \mid X) \,\|\, Q(H)) \,.$$

By the chain-rule decomposition established above, this quantity is exactly $\mathcal{C}_{\mathrm{SSIB}}$. Substituting this identity into the generic randomized-encoder generalization inequality yields (40). □

Proposition B.1 then implies that, when predictive risk is maintained, reducing $\mathcal{C}_{\mathrm{SSIB}}$ reduces an upper bound on the generalization gap of the induced stochastic predictor. This provides a formal explanation for why the SSIB objective can discourage over-informative memories that encode nuisance or spurious context features.

B.2 Robustness

We next provide a partial formal explanation for the robustness benefits of SSIB. The key idea is that input perturbations propagate through the recurrent memory only through the effective retention coefficients. Since Baleen regularizes the stochastic state transition mechanism, it contracts this perturbation propagation path.

Consider the SSM update

$$H_t = A_t \odot H_{t-1} + B_t \odot (1_N x_t^\top), \qquad t \in [T]. \tag{41}$$

Let $X' = [{x'_1}^\top, \ldots, {x'_T}^\top]^\top$ be a perturbed input sequence with

$$x'_t = x_t + \delta_t,$$

and let H'_t denote the corresponding hidden states generated by the same model on X':

$$H'_t = A'_t \odot H'_{t-1} + B'_t \odot (1_N {x'_t}^\top), \tag{42}$$

where $A'_t := A_t(X'_{\le t})$ and $B'_t := B_t(X'_{\le t})$.

Define the hidden-state perturbation

$$\Delta H_t := H'_t - H_t.$$

Then

$$\Delta H_t = A'_t \odot \Delta H_{t-1} + (A'_t - A_t) \odot H_{t-1} + (B'_t - B_t) \odot (1_N x_t^\top) + B'_t \odot (1_N \delta_t^\top). \tag{43}$$

We use this identity to upper-bound perturbation propagation.

Proposition B.2 (Perturbation propagation through stochastic memory)**.** *Assume there exists a matrix norm $\|\cdot\|$ and constants $L_A, L_B \ge 0$ such that for all t,*

$$\|A'_t - A_t\|_\infty \le L_A \|\delta_t\|, \tag{44}$$

$$\|B'_t - B_t\|_\infty \le L_B \|\delta_t\|, \tag{45}$$

and define

$$\rho_t := \max\left\{\|A_t\|_\infty, \|A'_t\|_\infty\right\}, \qquad c_t := L_A \|H_{t-1}\| + L_B \|1_N x_t^\top\| + \|B'_t\|_\infty \|1_N\|. \tag{46}$$

Then the hidden-state perturbation satisfies

$$\|\Delta H_t\| \;\le\; \rho_t \|\Delta H_{t-1}\| + c_t \|\delta_t\|, \tag{47}$$

and hence, by recursion,

$$\|\Delta H_t\| \;\le\; \sum_{s=1}^{t} \left(\prod_{j=s+1}^{t} \rho_j \right) c_s \|\delta_s\|. \tag{48}$$

Proof. Starting from (43), apply the triangle inequality and the bound

$$\|M \odot U\| \le \|M\|_\infty \|U\|$$

to each term:

$$\|\Delta H_t\| \le \|A'_t\|_\infty \|\Delta H_{t-1}\| + \|A'_t - A_t\|_\infty \|H_{t-1}\| \tag{49}$$

$$+ \|B'_t - B_t\|_\infty \|1_N x_t^\top\| + \|B'_t\|_\infty \|1_N \delta_t^\top\|. \tag{50}$$

Using the Lipschitz assumptions on A_t and B_t, and absorbing constants into c_t, yields (47). Repeated substitution gives (48). □

Proposition B.2 shows that the effect of a perturbation at time s on the hidden state at time t is damped by the multiplicative factor

$$\prod_{j=s+1}^{t} \rho_j.$$

Therefore, robustness improves when the model reduces the effective retention coefficients ρ_j, since this contracts the pathway along which perturbations accumulate over time.

The next corollary lifts the bound to the prediction space.

Corollary B.3 (Output perturbation bound)**.** *Assume the decoder g_Θ is L_g-Lipschitz with respect to the hidden state, i.e.,*

$$\|g_\Theta(H) - g_\Theta(\widetilde{H})\| \le L_g \|H - \widetilde{H}\|.$$

Then the output perturbation satisfies

$$\|\widehat{Y}'_t - \widehat{Y}_t\| \;\le\; L_g \sum_{s=1}^{t} \left(\prod_{j=s+1}^{t} \rho_j \right) c_s \|\delta_s\|. \tag{51}$$

Proof. This follows immediately from Proposition B.2 and the Lipschitz property of g_Θ. □

We now discuss the implications for Baleen. In Baleen-B, the KL term acts directly on the Bernoulli-distributed transition gates and encourages sparse retention probabilities. Hence the expected effective retention coefficient $\mathbb{E}[\rho_t]$ is reduced, which decreases the products in (48) and (51). In Baleen-E, the transition takes the form

$$A_t = \exp\big(A_t^\circ \operatorname{diag}(\Delta_t)\big), \qquad A_t^\circ < 0.$$

Therefore, larger timestep values Δ_t induce smaller transition magnitudes A_t and shorter effective memory time scales, again reducing the accumulation of perturbations from distant tokens. Consequently, the SSIB regularization can be interpreted as contracting the perturbation propagation path through the stochastic memory trajectory.

Remark. The analysis above is intentionally partial. It does not provide certified adversarial robustness, nor does it cover all nonlinearities in the full network. Rather, it formalizes the mechanism by which SSIB can improve robustness: by reducing effective retention of irrelevant or noisy history, it suppresses the long-range propagation of perturbations through recurrent memory.

B.3 Interpretation

We emphasize that our token-level score should be interpreted as a *mechanistic proxy* for *history-mediated influence*, rather than a universally faithful attribution score for the full nonlinear predictor. We now formalize this statement in a simplified setting.

Recall the token-level score

$$S_t := \frac{1}{ND} \sum_{n=1}^{N} \sum_{d=1}^{D} A_t^{(n,d)}, \tag{52}$$

where $A_t \in [0,1]^{N \times D}$ denotes the effective transition matrix at time t.

Fix a future time step $k \geq t$. Define the post-t retention factor

$$\Psi_{t+1:k}^{(n,d)} := \prod_{j=t+1}^{k} A_j^{(n,d)}. \tag{53}$$

The component of H_k that is inherited from the pre-t memory H_{t-1} through the history-retention path passing through A_t is

$$H_{k,\text{hist}}^{(t)} = \left(A_t \odot \Psi_{t+1:k}\right) \odot H_{t-1}. \tag{54}$$

Assume the logit of interest at step k is a linear readout of the hidden state:

$$z_k = \langle C_k, H_k \rangle = \sum_{n=1}^{N} \sum_{d=1}^{D} C_k^{(n,d)} H_k^{(n,d)}. \tag{55}$$

Then the contribution of the history-retention path associated with time t is

$$z_{k,\text{hist}}^{(t)} = \sum_{n=1}^{N} \sum_{d=1}^{D} C_k^{(n,d)} H_{t-1}^{(n,d)} \Psi_{t+1:k}^{(n,d)} A_t^{(n,d)}. \tag{56}$$

For convenience, define

$$w_{k,t}^{(n,d)} := C_k^{(n,d)} H_{t-1}^{(n,d)} \Psi_{t+1:k}^{(n,d)}. \tag{57}$$

Then (56) becomes

$$z_{k,\text{hist}}^{(t)} = \sum_{n=1}^{N} \sum_{d=1}^{D} w_{k,t}^{(n,d)} A_t^{(n,d)}. \tag{58}$$

Proposition B.4 (Average gate score controls history-mediated logit contribution)**.** *Assume that for some fixed future step $k \geq t$, the coefficients $w_{k,t}^{(n,d)}$ satisfy*

$$w_{k,t}^{(n,d)} \geq 0, \qquad \forall (n,d) \in [N] \times [D]. \tag{59}$$

Then the history-mediated logit contribution $z_{k,\text{hist}}^{(t)}$ is coordinate-wise nondecreasing in A_t, i.e.

$$\frac{\partial z_{k,\text{hist}}^{(t)}}{\partial A_t^{(n,d)}} = w_{k,t}^{(n,d)} \geq 0, \qquad \forall (n,d). \tag{60}$$

Moreover, if there exist constants $0 < m \leq M < \infty$ such that

$$m \leq w_{k,t}^{(n,d)} \leq M, \qquad \forall (n,d), \tag{61}$$

then

$$mND\,S_t \;\leq\; z_{k,\text{hist}}^{(t)} \;\leq\; MND\,S_t. \tag{62}$$

Consequently, under (59)–(61)*, a larger average-gate score S_t implies a larger history-mediated contribution to the future logit z_k.*

Proof. By (58),

$$z_{k,\text{hist}}^{(t)} = \sum_{n,d} w_{k,t}^{(n,d)} A_t^{(n,d)}.$$

Differentiating with respect to each coordinate gives

$$\frac{\partial z_{k,\text{hist}}^{(t)}}{\partial A_t^{(n,d)}} = w_{k,t}^{(n,d)},$$

which proves the coordinate-wise monotonicity under (59).

Next, using (61),

$$m \sum_{n,d} A_t^{(n,d)} \;\leq\; \sum_{n,d} w_{k,t}^{(n,d)} A_t^{(n,d)} \;\leq\; M \sum_{n,d} A_t^{(n,d)}.$$

Substituting (52), namely

$$\sum_{n,d} A_t^{(n,d)} = ND\,S_t,$$

yields (62). □

Table 5: Zero-shot accuracy (%) for ∼500M hybrid models trained on 10B FineWeb-Edu tokens (3:1 linear-to-attention). Best in **bold**, second underlined.

Method	ARC-C	ARC-E	BoolQ	GPQA	Hella.	MMLU	OBQA	PIQA	TruthfulQA	Wino.	Avg.
Pretrained ∼500M hybrid models on FineWeb-Edu (10B tokens)											
Mamba2	23.55	32.79	49.14	**25.67**	28.06	23.12	26.00	53.37	**25.21**	50.75	33.77
Baleen-B	24.32	45.58	57.37	24.55	33.99	22.97	29.60	62.68	23.87	50.99	37.59
Mamba3 SISO	24.80	44.00	55.80	25.30	32.80	23.10	29.20	60.80	24.20	51.10	37.11
Gated DeltaNet	**25.68**	46.38	**59.85**	24.55	34.61	22.83	**32.00**	62.79	24.36	51.22	38.43
Baleen-B + Mamba3 SISO	25.52	**47.33**	59.15	25.00	**35.24**	**23.46**	31.20	**63.45**	24.76	**51.79**	**38.69**

Proposition B.4 does not claim that S_t fully characterizes the total influence of token t on the final prediction. Rather, it shows that under a monotone linear-readout approximation, the average transition magnitude S_t is an order-preserving proxy for the *history-retention contribution* of token t to future logits. This gives a mechanistic interpretation to our score: larger S_t means that more of the pre-t memory is preserved and allowed to affect downstream predictions.

Remark. The proposition is especially natural for Baleen-B, where A_t directly plays the role of a stochastic retention gate. For Baleen-E, the same interpretation applies to the induced effective transition magnitude $A_t = \exp(A_t^{\circ} \operatorname{diag}(\Delta_t))$, although in that case the score should be viewed more cautiously as a proxy for state-update salience rather than a pure binary retain/drop quantity.

C Experiment Details

C.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.

D Additional Experiments

D.1 Language Benchmark with Hybrid Attention Models

We additionally evaluate four ∼500M-parameter hybrid models pretrained on 10B tokens of FineWeb-Edu, all using a Multi-Latent Attention (MLA) layer at every fourth position (a 3:1 linear-to-attention ratio). On a Mamba2 backbone, Baleen-B closes most of the gap to the strong Gated DeltaNet baseline (37.59% vs. 38.43% average) but does not fully match it. We attribute this to the recurrent layer itself: prior work has reported that Mamba2 is a suboptimal recurrent component for attention-hybrid stacks (Jamba Team et al., 2024). To test whether our SSIB objective transfers across SSM generations, we apply the same construction to the recently proposed Mamba3 SISO backbone (Lahoti et al., 2026). The resulting Baleen-B variant on Mamba3 reaches 38.69% on average—surpassing both its underlying Mamba3 baseline and Gated DeltaNet—and obtains the best score on five of the

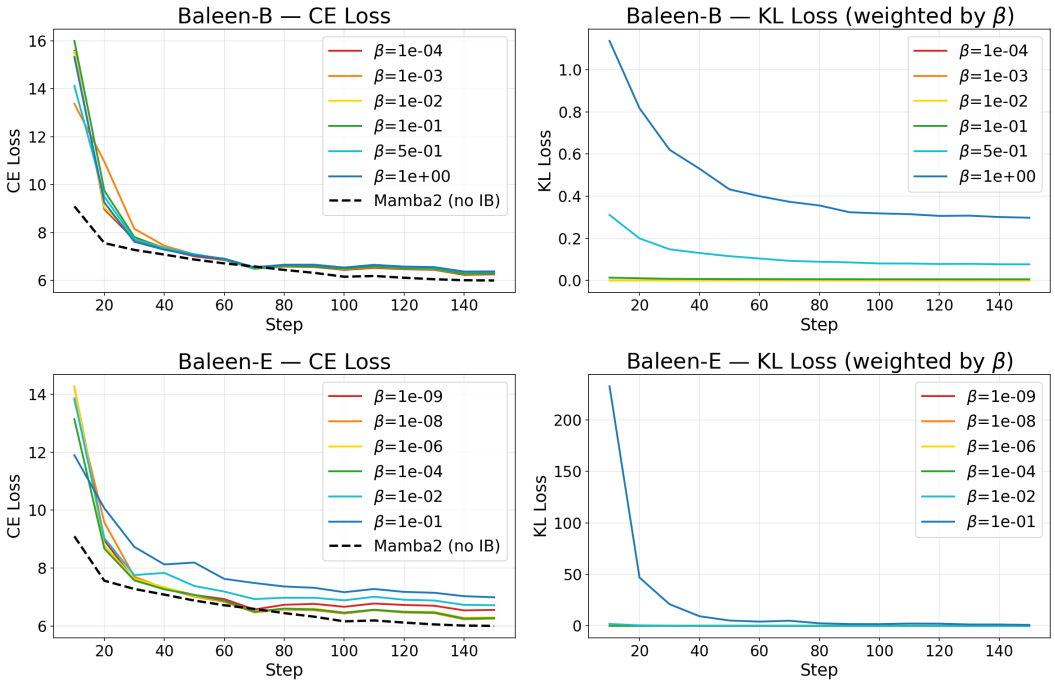


Figure 3: Training dynamics of Baleen-B (top) and Baleen-E (bottom) across different β values. **Left**: CE loss. For Baleen-B, all β converge to similar CE near the Mamba2 baseline, while Baleen-E is sensitive to β: large values ($\beta \geq 10^{-2}$) significantly slow convergence. **Right**: the β-weighted KL loss. For Baleen-B it decreases steadily, with larger β inducing higher but stable regularization; for Baleen-E it exhibits large initial magnitudes for aggressive β, reflecting over-regularization of the exponential bottleneck.

ten tasks (ARC-Easy, HellaSwag, MMLU, PIQA, Winogrande), confirming that our SSIB framework contributes orthogonal signal beyond what the choice of recurrent kernel provides. The remaining advantage of Gated DeltaNet over the Mamba2-based Baleen-B suggests that the delta-rule update is itself a factor making a recurrent backbone more amenable to attention hybridisation; encouragingly, Gated DeltaNet is fully compatible with our SSIB framework—a Bernoulli IB can be applied on top of its per-token update gate $\beta \in (0, 1)$ in a plug-in manner—and we leave a complete Baleen-B + DeltaNet to future work.

D.2 Ablations on β

We ablate the auxiliary loss weight β that controls the strength of the information bottleneck regularization. Figure 3 shows the training dynamics for Baleen-B (top row) and Baleen-E (bottom row). Baleen-B is robust to β: all configurations from 10^{-4} to 10^{0} converge to similar CE loss near the Mamba2 baseline, with the weighted KL term decreasing smoothly throughout training. This indicates that the Bernoulli bottleneck effectively self-regulates information flow without destabilizing optimization. In contrast, Baleen-E is sensitive to over-regularization. While moderate values ($\beta \leq 10^{-4}$) achieve CE loss comparable to the baseline, aggressive choices ($\beta \geq 10^{-2}$) induce excessively high KL penalties early in training, slowing convergence and preventing the model from recovering. This is attributable to the unbounded nature of the Exponential KL divergence, which amplifies the regularization effect relative to the bounded Bernoulli KL. Based on these results, we adopt $\beta = 10^{-1}$ for Baleen-B and $\beta = 10^{-6}$ for Baleen-E by default unless otherwise specified.

D.3 Empirical Analysis of Context Compression

To empirically verify that the information bottleneck objective achieves measurable context compression, we analyze the gating behavior and hidden-state geometry of our fine-tuned 7B models. We

Table 6: Compression metrics for fine-tuned 7B models. Baleen variants are measured at IB-designated layers (IB) and non-IB layers (NON-IB); Mamba2 is reported as a whole-network average. ↓ = lower is more compressed; ↑ = higher is more selective.

Model	Layers	Mean gate ↓	Sparsity (p<0.5) ↑	Gate std ↑	Eff. rank ↓
Mamba2	NON-IB	0.874	7.62%	0.178	418.3
Baleen-B	IB	**0.832**	**14.91%**	**0.297**	**378.0**
	NON-IB	0.876	6.66%	0.201	429.2
Baleen-E	IB	<u>0.851</u>	<u>12.92%</u>	<u>0.274</u>	**378.0**
	NON-IB	0.880	6.30%	0.198	429.4

compare Baleen-B and Baleen-E against the Mamba2 baseline, focusing on the three IB-designated layers (layers 0, $L/2-1$, $L-1$) where the bottleneck is applied. We evaluate on 200 sequences of length 2048 sampled from WikiText-2 (Merity et al., 2016) and report two complementary metrics: *gate statistics* characterizing the retention behavior of the state transition, and *effective rank* measuring the intrinsic dimensionality of the hidden representations.

Gate statistics. For every model and layer, we compute the retention probability $p_t^{(n,d)} = \exp\big(\texttt{softplus}(\boldsymbol{\Delta}_t + \mathbf{b}) \cdot \hat{\boldsymbol{A}}\big)$ and report the mean gate value, the fraction of gates below 0.5 (*sparsity*), and the standard deviation across all positions and heads. Results are summarized in Table 6.

Gate sparsity and selectivity. At IB layers, both Baleen variants exhibit markedly different gating behavior from Mamba2. Baleen-B achieves the highest sparsity (14.91% vs. 7.62% for Mamba2), nearly doubling the fraction of gates that fall below 0.5; Baleen-E shows a similar trend at 12.92%. This confirms that the KL regularization in the SSIB objective actively encourages the model to discard a larger fraction of past context. Meanwhile, the gate standard deviation at IB layers is substantially higher for both Baleen variants than the Mamba2 average (0.297 and 0.274 vs. 0.178), indicating that the information bottleneck promotes *selective* gating—the model learns to sharply discriminate between information worth preserving and information to discard, rather than forgetting uniformly.

Effective rank. We measure the effective rank (Roy & Vetterli, 2007) of hidden states at each layer, defined as $\mathrm{EffRank}(\mathbf{H}) = \exp\big(-\sum_i \hat{\sigma}_i \log \hat{\sigma}_i\big)$, where $\hat{\sigma}_i = \sigma_i / \sum_j \sigma_j$ are the normalized singular values of the centered hidden-state matrix $\mathbf{H} \in \mathbb{R}^{T \times D}$. Baleen IB layers achieve an effective rank of $\sim$378, substantially lower than the Mamba2 average of 418.3, indicating that the information bottleneck compresses the hidden representations into a lower-dimensional subspace. Notably, the non-IB layers of both Baleen variants remain virtually identical to Mamba2 ($\sim$429), confirming that the compression effect is well-localized to the designated IB layers and the bottleneck operates as a targeted, plug-in regularizer without distorting the broader network dynamics.