NORTHWESTERN
UNIVERSITY

Computer Science Department

Technical Report
Number: NU-CS-2026-30

June, 2026

SentenceSSM: Exploring Sentence as an Input Granularity for Selective State-Space Models

Jianwen Lyu

Abstract

State-space models such as Mamba decide what to keep in their hidden state through a content-based selection mechanism that normally operates on tokens. A token often carries only part of a semantic unit, so each selection decision rests on incomplete information. This work asks whether a more complete unit such as a sentence suits the mechanism better. We build SentenceSSM, which keeps a Mamba2 backbone unchanged and replaces its input unit with whole-sentence SONAR embeddings, paired with a jointly trained decoder, and compare it against a token-level Mamba2 at a matched parameter and token budget. Our central result is mechanistic. The selective gate concentrates its largest values on the a priori important sentences, with complete separation from the token-level model across all one hundred trajectories we test. We read this as a differential observation consistent with the granularity hypothesis, not as proof, and we name the control that would separate granularity from the pretrained encoder. We present this as a conceptual exploration rather than a settled result.

Keywords

State-Space Models, Mamba, Learning Representation, Continuous Representation Learning

NORTHWESTERN UNIVERSITY

SENTENCE-SSM: EXPLORING SENTENCE AS AN INPUT GRANULARITY FOR SELECTIVE STATE-SPACE MODELS

A THESIS
SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
for the degree

MASTER OF SCIENCE

Computer Science

By

Jianwen Lyu

EVANSTON, ILLINOIS

June 2026

Abstract

State-space models such as Mamba decide what to keep in their hidden state through a content-based selection mechanism, and that mechanism normally operates on tokens. A token often carries only part of a semantic unit, so the selection decision taken at each step rests on incomplete information that cannot be revised later. This thesis asks whether the token is the right input granularity for that mechanism, or whether a more complete unit such as a sentence would suit it better.

We instantiate the question as SentenceSSM. The model keeps a Mamba2 backbone unchanged and replaces its input unit with whole-sentence SONAR embeddings. A frozen SONAR encoder turns text into sentence vectors, the backbone predicts the next sentence vector, and a jointly trained SONAR decoder turns predictions back into text. We compare SentenceSSM against a token-level Mamba2 trained from scratch at a matched parameter and token budget, across sentence-level generation quality, general language ability, and a mechanistic probe of the selective gate.

Our central result is mechanistic. On importance-structured input the gate concentrates its largest selection values on the a priori important sentences in SentenceSSM, with complete separation from the token-level model across all one hundred trajectories we test. The behavioral comparison points the same way but is weaker, since both models are underfit at this budget and the jointly fine-tuned decoder confounds the text-level gains. We read the gate result as a differential observation consistent with the granularity hypothesis, not as proof of it. The separation has two candidate causes, the shift in granularity and the quality of the pretrained sentence encoder, and we name the control that would separate them. We present this work as a conceptual exploration and as a falsifiable starting point rather than a settled result.

Acknowledgment

I would like to thank my advisors, Qineng Wang, Zihan Wang, Manling Li, from MLL lab, Northwestern University, for the guidance and feedback that shaped this work, and for the freedom to pursue this question.

I also appreciate my committee members, Manling Li and David Demeter, for their time and for the questions that sharpened this thesis.

Preface

List of Abbreviations

Selectivity	The gate's per-unit decision of how strongly an input is written into the state.
Δt	The input-dependent step size setting that write strength. Large writes in, small skips.
Hidden state	The fixed-size recurrent memory carried in place of a KV cache.
Sentence embedding	A fixed-dimensional vector encoding a whole sentence, from the frozen SONAR encoder.
Trajectory	A contiguous sentence sequence from one document, packed in embedding space.
Chain sentence	A VAR = value sentence carrying the tracked information, important a priori.
Distractor sentence	A natural-language sentence that carries no tracked information.
Teacher forcing	Feeding the ground-truth input at each step rather than the model's own prediction.
Recency bias	The state's tendency to favor recent inputs as earlier ones decay with distance.
Decoder degradation	The decoder producing fluent text from its own prior while ignoring the backbone vector.
Joint training	Updating the Mamba backbone and SONAR decoder together so the decoder co-adapts.

Glossary

Selectivity	The gate's per-unit decision of how strongly an input is written into the state.
Δt	The input-dependent step size setting that write strength. Large writes in, small skips.
Hidden state	The fixed-size recurrent memory carried in place of a KV cache.
Sentence embedding	A fixed-dimensional vector encoding a whole sentence, from the frozen SONAR encoder.
Trajectory	A contiguous sentence sequence from one document, packed in embedding space.
Chain sentence	A VAR = value sentence carrying the tracked information, important a priori.
Distractor sentence	A natural-language sentence that carries no tracked information.
Teacher forcing	Feeding the ground-truth input at each step rather than the model's own prediction.
Recency bias	The state's tendency to favor recent inputs as earlier ones decay with distance.
Decoder degradation	The decoder producing fluent text from its own prior while ignoring the backbone vector.
Joint training	Updating the Mamba backbone and SONAR decoder together so the decoder co-adapts.

Table of Contents

List of Tables, Illustrations, Figures, and Graphs

Introduction

Selectivity is at the core of how State-Space Models (SSM) such as Mamba decide what to retain in their hidden states during training steps. This mechanism itself has been evaluated by several existing works [1, 2, 4, 6, 5, 7]. However, the default input granularity of Mamba is in most cases set to token, which is same as token-based Transformer model. Whether the token is in fact the unit on which the selectivity mechanism is best positioned to operate, and whether a sentence or concept-level representation would serve it better, remains an open question.

The selectivity mechanism of Mamba computes its gating parameters from a short causal convolution (width $d_{conv} = 4$), therefore the local context that determines each gating decision spans only a few neighboring positions [1]. When the input granularity is the token, each decision is therefore parameterized by at most 3 to 4 tokens within that convolution window, this is a span that captures only part of the surrounding semantics. Mamba moreover maintains no attention-based KV cache and no access to earlier tokens, so all prior context is compressed into a fixed-size recurrent state rather than retained for later reference [1]. As a result, by the time an entire semantic unit has been traversed, the decisions taken at its previous steps (based on partial semantics) have already been written into the hidden state. These decisions cannot be redone in light of information that arrives subsequently.

However, a sentence-level input can probably change the information available at the moment of gating: each unit on which selection operates is a sentence carrying comparatively more complete semantics. In this case, the gate may, when it acts, have a relatively more complete basis for its decision. We instantiate this idea as SentenceSSM (SSSM), a Mamba2 [2] backbone that consumes a stream of pretrained SONAR sentence embeddings [3], trained jointly with a sentence-to-text decoder so that the trained model produces text rather than embeddings.

Through SSSM, our aim is to explore whether shifting the input unit from the token to the sentence brings the input into closer correspondence with the granularity at which selectivity acts, and

whether such a shift leaves any behavioral or mechanistic trace at matched scale. We formulate this exploration as a controlled comparison against a standard token-level Mamba2, holding parameter count and pretraining-token budget fixed so that the two models are matched in scale, though as Section 3.1 notes they are not matched in their use of a pretrained sentence encoder.

The investigation proceeds in two steps, from behavior to mechanism. Behaviorally, both SSSM and the baseline are pretrained under a fixed budget. Within that limit, SSSM is modestly ahead on some general benchmarks (e.g., HellaSwag) and produces qualitatively more coherent decoded text. Mechanistically, on value-tracking samples we observe that SSSM's selectivity concentrates comparatively more on the key sentences, whereas the baseline spreads its emphasis more evenly and does not single them out. We observed clear patterns in the behavioral comparisons. Both models are confounded by a limited training budget, so we read the patterns rather than the exact scores. We therefore frame this work as a conceptual exploration rather than a systematic empirical study (details in Limitations section).

The main contributions of this work are:

- We raise the possibility that selective state-space models face a granularity mismatch, since a sentence may give the selection mechanism more complete semantic information to act on than a token does.
- We propose SSSM, a new architecture with a whole-sentence SONAR embedding as the input unit of a Mamba2 backbone, joined through input and output adapters to a jointly trained SONAR decoder so that the model is an end-to-end text generator. This fulfills the gap of existing works, where prior sentence-level models use a Transformer or recurrent-Transformer backbone and prior non-token SSMs operate at the byte level.
- We use SSSM as a mechanistic instrument. We introduce a gate-selectivity probe that reads the selective step size Δt on importance-structured input under teacher forcing, and we report a clean differential observation. Under sentence-level input the gate concentrates on the

a priori important sentences while a matched token-level model does not, with complete separation across all 100 trajectories we test (see details in Results section). We report this as a mechanistic observation consistent with the granularity hypothesis, not as proof of it. The observed difference has two candidate causes, the shift in granularity and the quality of the pretrained SONAR encoder. We name the control that would separate them in the Limitations section.

The remainder of this work is organized as follows. Section 2 reviews related work. Section 3 describes SentenceSSM and the training procedure. Section 4 details the experimental setup. Section 5 reports results. Section 6 synthesizes them against the hypothesis above. Section 7 discusses limitations. Section 8 concludes.

Related Work

Mamba and Content-Based Selection

Mamba's selection mechanism makes the state-update parameters functions of the current input, so that at each step the model can selectively propagate or forget information along the sequence depending on the current token [1]. This replaces the time-invariant dynamics of earlier structured state-space models (S4 and its relatives), in which the same update is applied regardless of content, with an input-dependent rule. The mechanism's intended role is thus a form of content-based selection: deciding (per step) how much of the current input to write into the recurrent state and how much of the existing state to retain.

Two synthetic tasks introduced alongside Mamba isolate this capability directly. On Selective Copying, where the model must copy a few content tokens while ignoring interspersed filler, the selective layer alone suffices to solve the task, whereas the earlier time-invariant SSMs fail even when combined with more expressive architectures [1]. On Induction Heads, an associative-recall task (having seen a bigram such as "Harry Potter," the model should complete "Harry" with "Potter"), Mamba solves the task by selectively retaining the relevant token while ignoring the intervening content, and extrapolates to sequences far longer than those seen in training [1]. Because these tasks are constructed so that only a small part of the input is relevant, success on them is evidence that the mechanism can keep what matters and discard the rest.

At the level of full language modeling, this selectivity translates into competitive quality: a 3B-parameter Mamba matches Transformers of twice its size on language modeling, in both pretraining and downstream evaluation, and is the first attention-free model to keep pace with a strong Transformer recipe as sequence length grows [1]. Independent in-context-learning studies report a similar pattern on synthetic algorithmic tasks, where Mamba learns sparse-parity problems that Transformers fail to learn [4], though the same work attributes part of this advantage to Mamba's

front-end causal convolution rather than to selection alone, and reports that Mamba in turn struggles on certain retrieval tasks.

Taken together, these results delineate what selection is and is not good at. Where the input has a clear importance structure, with a few relevant items embedded in irrelevant context, the mechanism can identify and preferentially retain the high-value content. This is the sense in which selectivity functions as a form of content-aware importance filtering. The qualification matters for what follows. This evidence concerns the mechanism's *ability to select*, demonstrated largely on synthetic tasks with a known relevance structure, and should not be read as a claim that Mamba is strong at copying or retrieval in general. That is a separate limitation we turn to next.

Limitations of Mamba

As demonstrated by existing works, Mamba's importance-aware selection does not help on every task. On several language tasks, pure Mamba still falls clearly behind Transformers, and two kinds of difficulty appear repeatedly in large-scale controlled comparisons.

The first difficulty is about in-context learning and answer formatting. Waleffe et al. [5] train 8B-parameter Mamba, Mamba-2, and Transformer models on identical data and find that, while pure SSMs match or exceed Transformers on many natural-language tasks, they fall behind on tasks that require strong in-context-learning or copying ability. On the standard 5-shot MMLU, at the 1.1T-token budget pure Mamba-2 scores about 17 points below the Transformer. This gap narrows to about 1.4 points when training is extended to 3.5T tokens. The authors attribute it mainly to two things: pure SSMs have trouble understanding the multiple-choice format, and they struggle to route each answer's knowledge into a single answer token. The gap is therefore not really about long-range retrieval. In fact, when the same task is posed in a cloze format, pure SSMs match or even exceed the Transformer, which suggests that this first difficulty is one that more training can largely close.

The second difficulty is more structural and does not dissolve with additional training. It concerns retrieving exact information from earlier in the context. On the synthetic Phonebook task, the Transformer answers with near-perfect accuracy up to its pretraining context length, whereas pure Mamba and Mamba-2 begin to fail once the input exceeds roughly 500 tokens, exhibiting a "fuzzy memory" in which the recalled value is close to but not the true one. For Mamba-2, this persists even at the 3.5T-token budget [5]. This pattern is consistent with a capacity argument. Jelassi et al. [6] prove that a two-layer Transformer can copy strings of exponential length, whereas generalized state-space models are fundamentally limited by their fixed-size latent state.

Existing work offers at least two distinct, mechanism-level accounts of why such retrieval is hard for SSMs. The first locates the bottleneck in the source of the gating decision. The selectivity parameters are computed from a short causal convolution, so each step can incorporate only a brief window of context around the current position, and semantics from further away cannot enter that step's importance judgment [1]. The second locates it in the decay of the hidden state. Wang et al. [7] show, both theoretically and empirically, that SSMs exhibit a structural *recency bias*, with the influence of one token on another decaying exponentially in their distance, which impairs recall of distant information. Over a long sequence, in other words, the limited state capacity is progressively consumed by intervening tokens, and information written early has largely decayed by the time the sequence ends.

Following this diagnosis, most proposed remedies pursue the same idea, namely restoring Mamba's ability to look back at context, and report tangible gains. These approaches fall into roughly three lines. The first interleaves a small number of attention layers into an otherwise SSM backbone, letting the model retrieve from the raw context directly [5, 4]. The second filters low-value state updates at inference time so that the limited memory concentrates on important content. LongMamba [8], for instance, first identifies critical tokens and then accumulates only those, mitigating the memory decay of the hidden state. The third improves the state-update rule itself. Gated DeltaNet [9] augments Mamba-2's gating with a delta rule, so that memory can be both erased

quickly and modified in a targeted, precise way.

What is worth noting is that, however different in mechanism, these three lines share a starting point. Each improves *how the existing token stream is used*, whether by looking back, by filtering tokens, or by refining the write rule. None of them asks whether the token stream is itself the most appropriate input granularity. From our view, the selection mechanism is designed for content-aware importance filtering, yet the input unit it faces at each step, the token, is often semantically incomplete. A semantically complete concept frequently spans several tokens, so the importance judgment made at a given step, which cannot be redone afterward, rests on locally incomplete semantics. We therefore advance a hypothesis that the token may not be the most suitable input granularity for this mechanism, and part of the gap observed on the tasks above may admit an explanation beyond the existing diagnoses. To our knowledge, discussion of Mamba's limitations has so far centered on restoring the model's opportunity to look back at context. Little prior work asks the converse question, whether a semantically more complete input representation could let the selection mechanism operate at a more appropriate granularity from the outset.

Alternative Input Granularities

Language models typically take the token as their smallest unit of processing, yet token boundaries are determined by statistical frequency (e.g., BPE) and do not align with semantic boundaries. A single concept often spans several tokens. Llama's tokenizer, for instance, splits "northeastern" into `['_n', 'ort', 'he', 'astern']`, none of which corresponds to a meaningful unit such as "north" or "east".

This mismatch is not merely cosmetic, and it affects the model's internal computation. Feucht et al. [10] observe a token "erasure" effect in Llama-2 and Llama-3. In the last-token representation of a multi-token word or named entity, information about the preceding and current tokens is rapidly forgotten in the early layers, a footprint of the model reassembling fragmentary tokens into

a word-level representation. Kaplan et al. [11] characterize this detokenization process further, showing that models recombine multi-token words (and even artificially split single-token words) into coherent word-level representations, and that this recombination takes place mainly in the early-to-middle layers. Their probes find that word and non-word representations become almost fully separable in the middle layers (peaking around 89%). The same study, however, also reports that about 22.6% of multi-token words are never decoded into the corresponding whole word at any internal layer, suggesting these words are not represented in the model's inner lexicon at all.

In other words, the semantic incompleteness of the token granularity is a structural phenomenon with internal evidence behind it. The model must first spend several layers assembling fragmentary tokens into complete semantic units before it can operate on them at a higher level, and this reassembly does not always succeed, computation that could be avoided if the input unit corresponded to a more complete semantic unit from the outset.

Faced with this, prior work has explored input granularities other than the token, in roughly two opposite directions.

One direction shrinks the granularity to bytes or characters, bypassing the tokenizer so the model builds representations from a more primitive signal. ByT5 [12] removes the tokenizer entirely and operates directly on UTF-8 bytes. Trained on roughly $4\times$ less text than a subword model, it remains competitive with a parameter-matched mT5, showing that tokenizer-free end-to-end modeling is feasible. The cost is that byte sequences are about $4\times$ longer than subword sequences (ranging across languages from roughly $2.5\times$ to $9\times$), which adds about 33% to pre-training time and makes inference roughly 4–$10\times$ slower on long-sequence tasks. Later work targets this bottleneck. BLT [13] uses a lightweight entropy model to set patch boundaries dynamically from the next-byte uncertainty, aggregating bytes into variable-length patches before the main backbone so that compute is concentrated where the data is more complex. It is the first tokenizer-free architecture to match tokenization-based models at scale. MambaByte [14] instead replaces the backbone itself, swapping the Transformer for the linear-complexity Mamba and thereby sidestepping the

quadratic-complexity bottleneck. Under a fixed parameter budget it reaches the loss of a byte-level Transformer in less than one-third of the compute, achieves lower bits-per-byte than MegaByte under matched compute, and retains natural robustness to noise and length extrapolation (to at least $4\times$ the training length). In our view the byte direction of MambaByte is aimed primarily at the statistical bias, brittleness, and inefficiency of tokenization, not at semantic completeness. Even so, the work shows that attaching a non-token granularity to a Mamba backbone is feasible and has precedent.

The other direction enlarges the granularity to the sentence, completing semantic integration before the input enters the model so that each input unit corresponds to a more complete semantic unit. Thought Gestalt (TG) [15] argues that token-level Transformers rely primarily on surface-level co-occurrence statistics and fail to form globally consistent representations of entities and events. It proposes a recurrent Transformer that processes one sentence at a time, compresses each sentence into a single "gestalt" vector held in a differentiable memory, and learns token and sentence representations jointly under a single next-token objective. Its scaling fits indicate that a matched GPT-2 needs roughly 5–8% more data and 33–42% more parameters to match TG's test loss, evidence that sentence-level processing carries independent value. SONAR [3] provides a concrete realization of a sentence-level representation, encoding a whole sentence into a fixed-dimensional vector rather than retaining its internal token sequence. SONAR was designed for multilingual, cross-modal retrieval rather than for the token-incompleteness problem, but it furnishes, as a by-product, a large-scale, high-quality sentence-level semantic space. LCM [16] then demonstrates the feasibility of using such sentence embeddings as model input. It performs autoregressive sentence prediction in the SONAR embedding space in place of a token sequence and reaches competitive results on generation tasks such as summarization, at scales up to 1.6B (later 7B) parameters. Sequence modeling at the sentence granularity is therefore viable not only at the representation level but on full language-generation tasks.

Taken together, these works reveal a clear gap. Combining a byte granularity with Mamba has

been done (MambaByte), whereas the attempts to raise the input granularity from the token *to the sentence* have so far all used a Transformer or recurrent-Transformer backbone (TG, LCM). No prior work brings this idea to an SSM architecture. Given the diagnosis in the preceding subsection, that the semantic incompleteness of the token granularity may be one root reason the selection mechanism cannot do better, combining sentence-level input with an SSM backbone is a natural yet so-far unexplored direction. This is where our work is situated.

Method

In this work we investigate whether sentence granularity is more suitable for Mamba's selective scan mechanism. To make that question testable we build a single end-to-end model, SentenceSSM (SSSM), that consumes sequences of sentence embeddings rather than tokens and propagates them through a Mamba2 backbone. We first describe the architecture and the training objective together with the trajectory format it operates on. We then lay out the experimental design, which organizes the evaluation into three stages (sentence-level generation quality, general language ability, and a mechanistic gate-selectivity probe) against a budget-matched token-level baseline. Finally we describe the training strategy and its freeze variants and the decoding paths together with the control that rules out a degenerate language-model explanation.

Model Construction

SSSM keeps Mamba's selective SSM backbone entirely unchanged and replaces only the unit on which that backbone operates, from token embeddings to whole-sentence embeddings, together with the input- and output-side adapters this substitution requires. The parameterization of the selection mechanism is left untouched, so the two models share an identical selective scan. They differ in two respects that the design does not separate. One is the input granularity. The other is the use of a pretrained SONAR encoder and decoder that the token-level baseline does not have. The token baseline learns its representations from scratch, whereas SSSM reads and writes through a sentence space pretrained on large multilingual data, so part of any difference we later observe may come from that pretraining rather than from granularity. We therefore do not treat granularity as the sole explanation, and we return to this confound, and to the control that would isolate it from encoder quality, in the Analysis and Limitations sections.

End-to-end data flow. Given a passage of text, a SONAR encoder [3], kept frozen throughout, first segments and encodes it into a sequence of sentence embeddings $X = (x_1, \ldots, x_S)$, where each

$x_t \in \mathbb{R}^{1024}$ encodes one complete sentence. This sequence is projected to the backbone width by an input adapter and passed to the Mamba backbone, which runs step by step over the sentence representations. The backbone output is projected back to 1024 dimensions by an output adapter, giving the prediction of the next sentence embedding, which a likewise frozen SONAR decoder maps back to text. Throughout this pipeline the SONAR encoder and decoder remain frozen, and the input and output adapters are the only learnable bridge between the SONAR representation space and the SSM backbone.

Changes relative to vanilla Mamba. Table 1 contrasts the two along five dimensions. In essence, the Mamba blocks themselves are unchanged. What changes is the object they operate on, namely sentence representations rather than token embeddings, together with the adaptation at the two ends.

Dimension	**Vanilla Mamba**	**SSSM**
Input	token ids via embedding lookup	precomputed SONAR sentence vectors $[B, S, 1024]$
Adapter	none	input adapter = LayerNorm + Linear$(1024 \to d_{\text{model}})$
Backbone	Mamba blocks over tokens	Mamba blocks over sentence representations
Output	vocabulary logits $[B, L, V]$	output adapter $(d_{\text{model}} \to 1024)$ predicting the next sentence vector
Decoding	argmax over the vocabulary	frozen SONAR decoder: sentence vector $\to$ text

Table 1: Architectural changes from vanilla Mamba to SSSM.

Formalization. Let $x_t \in \mathbb{R}^{1024}$ be the input sentence embedding at step t. The input adapter normalizes and then linearly projects it:

$$\tilde{x}_t = W_{\text{in}} \, \text{LN}(x_t).$$

The backbone is Mamba's selective SSM unchanged. Its continuous parameters (A, B) are discretized into $(\bar{A}_t, \bar{B}_t)$ through an input-dependent step size Δ_t, and the state is updated and read out as

$$\bar{A}_t = \exp(\Delta_t A), \qquad \bar{B}_t = (\Delta_t A)^{-1}(\exp(\Delta_t A) - I)\,\Delta_t B,$$

$$h_t = \bar{A}_t\, h_{t-1} + \bar{B}_t\, \tilde{x}_t, \qquad y_t = C_t\, h_t.$$

These discretization and update equations restate the standard Mamba and Mamba2 formulation [1, 2], reproduced here only to define the step size Δ_t that the gate-selectivity probe of Section 3.5 inspects, and they are not a contribution of this work. The step size Δ_t and the matrices B and C are still computed from the current input $\tilde{x}_t$ exactly as in Mamba, and we leave their parameterization unchanged. (In Mamba2, Δ_t is a per-channel vector rather than a single scalar. We write it as a scalar in the discretization above for brevity, and in Section 3.5 we probe its mean across the state dimension.) Δ_t is the quantity through which the selection mechanism sets how strongly the current input is written into the state, and it is the quantity probed in Section 3.5. The output adapter then projects the backbone output back into the sentence-vector space to predict the next sentence embedding:

$$\hat{x}_{t+1} = W_{\text{out}}\, \text{norm}_f(y_t) \in \mathbb{R}^{1024}.$$

Relation to existing sentence-level models. SSSM predicts the next sentence embedding autoregressively in the SONAR sentence-vector space, which it shares with LCM (Section 2). It differs in two respects. First, the backbone is a selective SSM rather than a Transformer. Second, and more importantly, the aim of this work is not to obtain stronger generation at the sentence granularity, but to examine the behavior of the selection mechanism itself under sentence-level input. The latter rests on the input-dependent selective step size Δ_t in the backbone, which a Transformer backbone does not provide.

Training Objective

Trajectory packing. A training example is a single trajectory, that is, a contiguous sequence of sentences from a source document, packed in embedding space. The pretraining that produces our primary checkpoint uses only the sentence embeddings of each trajectory, packed as a continuous stream terminated by an EOT ("end of trajectory") marker, with several trajectories concatenated to fill the fixed sequence length:

$$[x_1, x_2, \ldots, x_n, \text{EOT}, x'_1, \ldots, x'_m, \text{EOT}, \ldots]$$

Each x_i is one 1024-dimensional SONAR sentence embedding, and the model predicts x_{i+1} from $x_{\leq i}$ at every sentence position. The prediction loss below is applied only at these sentence positions, never on the EOT markers or on padding, so that gradient flows where the model predicts a next sentence. Padding to the fixed length is performed in embedding space with a designated padding vector and is excluded from both attention and loss. The maximum trajectory length in our 780M experiments is 64 sentences, each bounded by a maximum of 512 subword tokens on the SONAR side.

MSE control variant. Because SSSM operates entirely in SONAR's sentence-vector space, the most direct training signal is a mean-squared error (MSE) in that space between the predicted next-sentence embedding $\hat{x}_{t+1}$ and the ground-truth next-sentence embedding x_{t+1}. We keep this MSE only as a control variant, because it optimizes the geometric placement of the predicted vector alone and we find embedding-space proximity to be a loose proxy for decoded-text quality. Its exact form is given in the Auxiliary Objective and Packing Format appendix.

Cross-entropy objective. The primary checkpoint is therefore trained not on MSE but on a token-level cross-entropy computed through the SONAR decoder, following the joint sentence-prediction and decoding objective of SONAR-LLM [17]. Let $\hat{x}_{t+1}$ be teacher-forced through the decoder D against the reference text $y_{t+1} = (w_{t+1,1}, \ldots, w_{t+1,L_t})$, yielding the per-token conditional probabilities $p\big(w_{t+1,i} \mid \hat{x}_{t+1}, w_{t+1,<i}\big)$. The objective is the mean cross-entropy over all tokens at the

sentence positions,

$$\mathscr{L}_{\mathrm{CE}} = -\frac{1}{\sum_t L_t} \sum_{t=1}^{T-1} \sum_{i=1}^{L_t} \log p\big(w_{t+1,i} \mid \hat{x}_{t+1}, w_{t+1,<i}\big).$$

The two objectives are alternatives, not a weighted sum. The primary model is trained with $\mathscr{L}_{\mathrm{CE}}$, while an otherwise-identical model trained on the MSE objective alone serves as a control variant. Under $\mathscr{L}_{\mathrm{CE}}$, gradients flow through $\hat{x}_{t+1}$ both into the backbone and into the SONAR decoder, so the decoder is itself fine-tuned. This is what makes the model a self-contained text generator. It also means that every text-level result reported later reflects the combined effect of the backbone and the fine-tuned decoder, not of sentence-level granularity in isolation, and we bound the interpretation accordingly in the Limitations section. Finally, because the model does not emit tokens directly, perplexity is not a natural training-time quantity. We use the cross-entropy checkpoint loss as the primary training metric and recover token-level perplexity only at evaluation time, through the SONAR decoder.

Experimental Design

Our evaluation is a three-stage progression against a single budget-matched baseline, with each stage asking a more demanding question than the last. Can SSSM *generate* coherent text, does sentence granularity *preserve* general language ability, and is the selective gate *actually using* the sentence context? We fix only the design here and defer all numerical results to Section 5.

Baseline. Throughout, the comparison is against a token-level Mamba2 language model trained from scratch under a matching 780M-parameter budget and a matching 14.4B-token budget on the same FineWeb-20B corpus. The parameter budget is matched on the trainable backbone. The frozen SONAR encoder and decoder that SSSM additionally uses are pretrained and are not counted toward the 780M, a point we return to as a confound in the Analysis and Limitations sections. Because the baseline is token-level and incompatible with the SONAR sentence pipeline, we

score each model through its own native path rather than forcing one architecture onto the other's pipeline, which would unfairly disadvantage the foreign side.

Stage 1 (sentence-level generation quality). On ROC-Stories, a clean five-sentence narrative benchmark disjoint from FineWeb, we compare decoded text on ROUGE-1, ROUGE-2, ROUGE-L, BLEU, sentence-level CoLA acceptability [18], and a 4-gram repetition rate. Each model decodes on its own native path. SSSM produces predicted sentence embeddings that are decoded by the fine-tuned SONAR decoder, while the baseline takes a token-level teacher-forced argmax decode in a single forward pass. To equalize the sample size and the reference text, all cross-architecture metrics are reported on the 1,477-trajectory subset on which the two runs share an identical ground truth (Section 5.2).

Stage 2 (general language ability). To check that moving to sentence granularity does not sacrifice broad competence, we evaluate on the full EleutherAI LM-Evaluation-Harness 0-shot suite: ARC-Easy, ARC-Challenge [19], HellaSwag [20], PIQA [21], and WinoGrande [22]. The baseline uses the standard harness. SSSM is scored through an evaluation adapter that feeds the context as sentence embeddings, predicts the next sentence embedding, and uses the fine-tuned SONAR decoder to score the target token sequence. The adapter can return either the raw sum of subword log probabilities or their per-subword mean, and only the raw-sum setting is comparable to the standard harness.

Stage 3 (mechanistic gate-selectivity probe). The central stage asks whether the selective gate is content-driven at the sentence level. We use the RULER variable-tracking (VT) task not as a performance benchmark, but because it supplies, by construction, sentences whose importance is known a priori. (In Section 1 the same task appears in the opposite role, as one on which retrieval-style behavior cannot be attributed to granularity alone.) A VT trajectory contains a few "chain" sentences of the form `VAR _ = _` that carry the tracked value, among natural-language "distractor" sentences that do not. During a single TF forward pass we extract the input-dependent step size Δt from each of the 48 Mamba2 blocks. For an N-sentence trajectory, each block yields one Δt value

per sentence and per state dimension, consistent with the per-channel nature of Δ_t (Section 3.1). We summarize each sentence by the layer-wise mean of Δt across the state dimension. In SSSM each sentence is already a single input position, so this value is read off directly. In the token-level baseline a sentence spans many tokens, so we align the two architectures explicitly. We segment each trajectory into sentences with the same SaT segmenter, map every token to its sentence by character-span alignment while excluding special tokens, take the channel mean of Δt at each token, and average these token values over the tokens of a sentence to obtain one sentence-level value. The summary used throughout is this mean over a sentence's tokens, the same mean-gate statistic computed for SSSM. Per layer we then record the rank of each chain sentence within the trajectory's Δt distribution, together with a chain-z-score, defined as the chain mean minus the distractor mean, divided by the distractor standard deviation. This identical pipeline is run on both models as a cross-architecture control, aggregated over 100 VT samples (Section 5.4).

Training Strategy

SSSM couples a trainable SSM backbone with a SONAR decoder, and which of the two is updated is itself a design choice. We adopt the joint regime for the primary checkpoint, training the backbone and the SONAR decoder at the same time, each with its own learning rate, so that the decoder can co-adapt to the backbone's output distribution in SONAR space. This choice rests on a qualitative analysis together with preliminary observations made during training, not on a controlled comparison, and the matched ablation that would settle it is left to future work. The two regimes we set aside, a fixed-decoder regime and a two-stage regime, together with the reasoning behind the choice, are detailed in the Training Hyperparameters and Regimes appendix (Table 7).

Decoding Paths and the Degenerate-LM Control

Because SSSM separates sentence modeling from token decoding, several distinct paths run from a context to generated text.

Teacher-forced (TF) decoding. At each position t we feed the ground-truth sentence embedding x_t as input and read out the prediction $\hat{x}_{t+1}$. We then send $\hat{x}_{t+1}$ through the fine-tuned SONAR decoder with the ground-truth target text y_{t+1} as the decoder's target sequence. The per-token cross-entropy, summed and exponentiated over a corpus, gives a teacher-forced perplexity. The argmax of the decoder logits gives a teacher-forced argmax decode, which is what Stage 1 uses for cross-architecture comparison.

Autoregressive (AR) decoding. Starting from a real context $x_1, \ldots, x_M$, we predict $\hat{x}_{M+1}$, send it through the SONAR decoder to recover text $\hat{y}_{M+1}$, re-encode $\hat{y}_{M+1}$ to obtain a clean sentence embedding x'_{M+1}, and condition on x'_{M+1} to predict $\hat{x}_{M+2}$. AR is the path used at generation time. Each AR step routes the predicted vector through the SONAR decoder and back through the encoder, so a sentence-to-token decoding bottleneck and the drift from re-encoding the model's own output can accumulate across steps in a way that teacher forcing avoids. The gate-selectivity probe of Stage 3 is therefore performed exclusively in the TF setting, which keeps the input distribution clean and ensures that any observed selectivity is a property of the trained model rather than of accumulated AR error.

Excluding a degenerate language-model explanation. A sentence-level model could in principle produce reasonable text while ignoring its sentence input, behaving as a plain token-level language model driven only by the decoder's prior. We rule this out with a ladder of three single-forward configurations that share the same trajectories and the same decoder, ordered by how much information they carry about the target sentence. The first feeds the gold SONAR encoding of the target, an encode-then-decode round-trip with no backbone, which is the upper bound. The second feeds the SSM's predicted vector, that is, the model itself under teacher forcing. The third removes the decoder's access to the sentence embedding, leaving only its language prior, which is a no-signal lower bound. Were SSSM a degenerate language model, all three would collapse toward the no-signal configuration. Instead, as Section 5.3 reports, the model sits well below the no-signal bound, so its predicted vector carries genuine sentence information. The autoregressive variant of

the second configuration is not used in this control, for the input-distribution reasons above.

Experimental Setup

Datasets

Pretraining corpus. A 14.4-billion-token slice of FineWeb-20B. Raw documents are sentence-segmented with wtpsplit (SaT), embedded into 1024-dimensional SONAR vectors, and precomputed and stored together with the source text, so the decoder can be fine-tuned at training time.

Evaluation sets. Stage 1 uses ROC-Stories [23], a five-sentence narrative benchmark disjoint from FineWeb. The SSSM and baseline runs are intersected by ground truth to a 1,477-trajectory matched subset used for cross-architecture metrics (Section 3.3). The precise split sizes and the per-run trajectory counts are given in the Dataset Splits appendix. Stage 2 uses the EleutherAI LM-Evaluation-Harness 0-shot suite (ARC-Easy, ARC-Challenge [19], HellaSwag [20], PIQA [21], and WinoGrande [22]). Stage 3 uses the RULER variable-tracking task (vt_4k) [24], restricted to 100 evaluation samples, each a long passage with a few `VAR _ = _` chain sentences among natural-language distractors.

Training Configuration

The 780M-parameter SSSM is trained on a single node with four A100 GPUs under a warmup-stable-decay schedule, to a 14.4B-token budget matched with the baseline. The full hyperparameter list, namely batch size, gradient accumulation, trajectory length, optimizer, and the reported token milestones, is given in the Training Hyperparameters and Regimes appendix. The primary SSSM checkpoint analyzed here is reached at 14.4B cumulative training tokens with a cross-entropy training loss of 3.1085. Its identifier, together with that of the budget-matched token-level baseline, is listed in the Implementation and Reproducibility appendix.

Results

We report results in three parts, following the experimental design of Section 3.3. Section 5.2 covers behavioral evaluation, namely sentence-level generation quality and general language ability. Section 5.3 reports the control that isolates the contribution of the selective scan. Section 5.4 reports the gate-selectivity probe. All SSSM numbers are for the 780M joint checkpoint at 14.4B tokens, and the baseline is the budget-matched token-level Mamba2 of Section 3.3.

Behavioral Evaluation

Generation quality. Table 2 reports sentence-level generation quality on the matched ROC-Stories subset. SSSM is ahead on the overlap metrics (ROUGE and BLEU) and on CoLA acceptability, where the margin is largest, 0.856 against 0.256, which indicates that its decoded sentences are far more often judged grammatical. The single reverse is the 4-gram repetition rate, where SSSM repeats more than the baseline, 0.099 against 0.003, which we tentatively attribute to collapse in the autoregressive re-encoding loop, where decoding a predicted vector to text and re-encoding it can settle into a repeated sentence. Table 3 shows a representative example. SSSM produces coherent on-topic sentences, while the token-level baseline, decoded on its own native path, degenerates into fragments. This qualitative gap is consistent with the large difference in CoLA acceptability.

General language ability. Table 4 reports the zero-shot LM-Evaluation-Harness suite. At the 14.4B-token budget both models sit close to chance on the multiple-choice tasks. On raw accuracy SSSM is modestly ahead on ARC-Challenge, HellaSwag, and PIQA, and even with the baseline on ARC-Easy and WinoGrande, while under length-normalized accuracy the two are roughly comparable. This indicates that moving the input unit from the token to the sentence does not degrade general language ability at matched scale.

Selective-Scan Controls

Metric	SSSM	Baseline
ROUGE-1 (F)	**0.241**	0.211
ROUGE-2 (F)	**0.070**	0.021
ROUGE-L (F)	**0.225**	0.194
BLEU (corpus)	**4.41**	0.90
CoLA acceptability	**0.856**	0.256
rep-4 (↓)	0.099	**0.003**

Table 2: Sentence-level generation quality on the 1,477-trajectory matched subset of ROC-Stories. Higher is better, except for rep-4, where lower is better. The token-level baseline is decoded on its own native single-forward teacher-forced argmax path. The best value in each row is in bold.

Ground truth	SSSM	Baseline
A New England family went out to Florida for vacation.	The family went to New York, where they spent the rest of the summer.	The new York- has to to the to the to
Hoping to get away from the cold, they marveled at warm temperatures.	The weather was cold and the sun was shining.	The, the the a, the local, the wereed the the,.
The car left Tallahassee and the family worked their way to the keys.	The family went to work.	They winter was a andasco, the city of to way to the local.
They made stops along the way, such as Orlando to enjoy the sights.	They had to wait for the sun to come out.	They were a. the way. and as the and the the summer.
Eventually, they made their way to their destination on Key Largo.	They traveled to California and then to New Mexico.	the city the way to the home. the

Table 3: Decoded text on one ROC-Stories example (trajectory 6123), comparing the ground-truth continuation, SSSM, and the token-level baseline on its native path. Each row is one sentence position. SSSM forms coherent on-topic sentences, while the baseline degenerates into fragments.

	SSSM		Baseline		
Task	acc	acc_norm	acc	acc_norm	**chance**
ARC-Easy	0.346	0.315	0.346	0.332	0.25
ARC-Challenge	0.216	0.252	0.190	0.220	0.25
HellaSwag	0.279	0.247	0.264	0.253	0.25
PIQA	0.562	0.529	0.552	0.546	0.50
WinoGrande	0.496	n/a	0.500	n/a	0.50

Table 4: Zero-shot LM-Evaluation-Harness accuracy, SSSM versus the budget-matched token-level baseline. Both models are near chance at the 14.4B-token budget. WinoGrande reports accuracy only.

To check that SSSM's generation reflects genuine use of its sentence input rather than a strong decoder prior, we run a decoding-path control on the primary checkpoint. It places the model's predicted vector between an encode-then-decode upper bound, which feeds the gold SONAR encoding of the target, and a no-signal lower bound, which removes the decoder's access to the sentence embedding and leaves only its language prior. Table 5 reports the teacher-forced perplexities. The model sits well below the no-signal bound, by nearly a factor of three, so its predicted vector carries genuine sentence information and the pipeline is not a disguised token-level language model.

Configuration	**Teacher-forced perplexity**
Gold SONAR encoding (upper bound)	36.5
SSSM predicted vector (the model)	62,732
No sentence signal (decoder prior only)	172,009

Table 5: Decoding-path control on the primary checkpoint, as teacher-forced perplexity (lower is better). The model's predicted vector lands well below the no-signal lower bound, which shows that the vector carries real sentence information rather than letting the decoder's language prior produce the text.

Gate Selectivity

Table 6 reports the gate-selectivity probe on RULER variable-tracking. The contrast is sharp. In SSSM the chain sentences, which carry the tracked values, sit at the top of the per-trajectory Δt distribution, with a mean rank of about 5 out of roughly 710 sentences and a mean z-score of $+7.11$, and every chain sentence falls in the top 12. In the token-level baseline the same sentences are pushed to the bottom, with a mean rank of about 636 and a mean z-score of -1.68, and none reach the top 12. The distractor sentences stay near the trajectory mean in both models, with mean z-scores of -0.09 for SSSM and $+0.007$ for the baseline. This separation is not an averaging artifact. On a per-sample paired comparison across the 100 trajectories, SSSM assigns the chain sentences a lower (better) rank than the baseline in every single sample, 100 of 100, with a paired Wilcoxon signed-rank $p \approx 4 \times 10^{-18}$ and a Cliff's δ of 1.0, the value that marks complete separation between the two models. The chain-z-score returns the identical verdict. Figure 1 plots the full per-sample distributions, which do not overlap between the two models on either statistic. Single-sample per-layer heatmaps in the Supporting Figures appendix, Figure 2 and Figure 3, show the effect directly, with the SSSM gate concentrating on the chain sentences while the baseline gate does not.

Metric	**SSSM**	**Baseline**
Chain rank, mean (↓)	**4.76**	635.67
Chain rank, median	**5.0**	706.0
Chain in top-12 (%)	**100.0**	0.0
Chain in top-50 (%)	**100.0**	0.83
Chain z-score, mean	$+7.11$	-1.68
Distractor z-score, mean	-0.09	$+0.007$

Table 6: Gate selectivity on RULER variable-tracking, aggregated over 100 samples (482 chain sentences, about 710 sentences per trajectory). Rank is within each trajectory's Δt distribution, where a lower rank means the sentence is more strongly written into the state. On a per-sample paired comparison SSSM ranks the chain sentences below the baseline in every sample, 100 of 100, with a paired Wilcoxon signed-rank $p \approx 4 \times 10^{-18}$ and Cliff's $\delta = 1.0$ for complete separation. The chain-rank and chain-z-score tests agree.

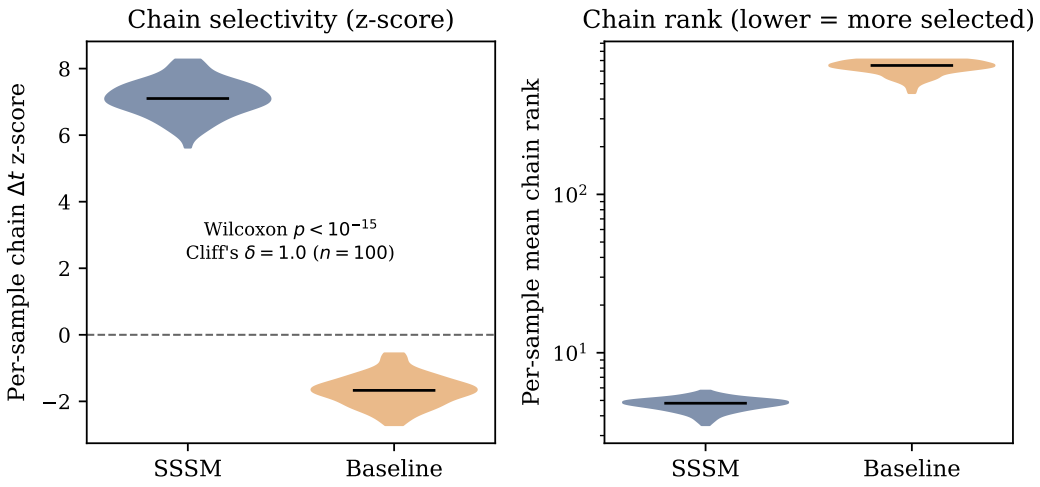

Figure 1: Per-sample distribution of chain-sentence selectivity over the 100 RULER variable-tracking trajectories, SSSM against the token-level baseline. The left panel shows the chain z-score on a linear axis, with the dashed line at zero marking the trajectory mean. The right panel shows the mean chain rank on a log axis, where a lower rank means stronger selection. The two models do not overlap on either statistic. A per-sample paired test confirms the gap, with a Wilcoxon signed-rank $p \approx 4 \times 10^{-18}$, a Cliff's δ of 1.0, and SSSM more selective in 100 of 100 samples.

Analysis

Through the results of Section 5 we evaluate the hypothesis of Section 1, that the token may be too fine a unit for the selective scan and that a sentence-level input brings the gating decision closer to the granularity at which selection acts. The evidence is of two kinds, behavioral and mechanistic, and they are not equally strong. We treat the mechanistic finding as the primary evidence and the behavioral comparison as a weaker corroboration, for the reasons made explicit below.

Behavioral evidence is positive but confounded. On ROC-Stories (Table 2) SSSM is ahead of the budget-matched token baseline on the overlap metrics and on grammatical acceptability, where the gap is largest. On the zero-shot suite (Table 4) the two models are close to chance at this 14.4B-token budget, with SSSM modestly ahead on raw accuracy and roughly even under length-normalized accuracy. Together, these indicate that transferring the input unit from the token to the sentence does not cost general competence and coincides with higher decoded-text quality on this confounded comparison. What's worth mentioning is that we do not define this as decisive

evidence. As we mentioned before, the SONAR decoder is fine-tuned together with the backbone, so the text-quality gap partially reflects the decoder rather than the input granularity, and under the insufficient training budget the benchmark numbers are immature. One number even cuts the other way, since SSSM repeats more on the 4-gram repetition rate.

The mechanistic evidence is the strongest part, though not confound-free. The gate-selectivity probe (Table 6) is the most direct test of the hypothesis, because it inspects the selection statistic Δt under teacher forcing and does not depend on decoded-text quality. As reported in Section 5.4, under sentence-level input the gate concentrates its largest Δt on the a priori important chain sentences, the ones carrying variable-tracking information, placing them at the top of the per-trajectory distribution, whereas the same mechanism on a token stream pushes them to the bottom, with the distractor sentences near the trajectory mean in both models. The separation holds per sample across all 100 trajectories rather than only in the mean, with complete separation between the two models (Figure 1). This pattern is consistent with the granularity hypothesis, and it connects back to the diagnosis of Section 2, that a gate parameterized from a short window over tokens cannot see a whole semantic unit at the moment it decides, while a sentence input gives it a more complete basis. One qualification belongs here rather than only in the Limitations, since it bears on the importance claim directly. The chain sentences are distributionally distinct from the natural-language distractors, because a `VAR _ = _` line is unlike ordinary prose, so part of the separation may reflect how separable that pattern is in SONAR space rather than a learned ranking of importance, and Section 7 states the task design that would test this. We read the result as a differential observation rather than as evidence that the gate has learned a better importance filter, for this reason and for the asymmetry developed next.

The contrast is asymmetric, and only one side is confounded. The rank separation between SSSM and the token model is asymmetric in what could explain it. It has two candidate causes. One is the shift in granularity. The other is the quality of the pretrained SONAR encoder that SSSM reads through. The token model never uses that encoder, so the encoder-quality argument

cannot explain its failure to concentrate. We therefore read the token side as not attributable to encoder quality. Whether it is driven by granularity or by the limited training budget stays open. The SSSM side stays open until the encoder control of Section 7 is run.

A control rules out a trivial explanation. A natural objection is that the predicted vector might carry little real information and that the decoder alone produces the text, in which case the Δt pattern would be an artifact of a disguised token-level language model. The decoding-path control on the primary checkpoint (Section 5.3, Table 5) addresses this. The model's predicted vector sits below the no-signal lower bound by nearly a factor of three, so it carries genuine sentence information and the pipeline is not a disguised token-level language model. The predicted vector nonetheless remains far above the encode-then-decode upper bound, so it is informative without being a near-perfect reconstruction, an absolute level consistent with the underfit regime of Section 7 rather than with a converged model. This is what licenses reading the Δt result as a property of sentence-level prediction rather than of the decoder.

Triangulation. The two lines of evidence point the same way. The behavioral comparison indicates that sentence granularity is at least not harmful and is probably mildly helpful, and the mechanistic probe shows a separation consistent with the hypothesis. The observation rests on the mechanistic result, because it is measured directly on Δt, is run under teacher forcing with a clean input distribution, and survives the decoder control above. Neither line, on its own, separates the effect of the input granularity from that of the pretrained sentence encoder, and we make that open question explicit in Section 7.

What the results do not show. The contribution here is an existence result at the mechanistic level, not a demonstration of stronger generation. The probe shows where the gate writes most strongly, not that SSSM tracks the queried variable more accurately, so the selectivity is not yet linked to task performance. At the budget studied the benchmark numbers are near chance, and the probe is aggregated over a single task family. We take up these limits, and the experiments that would close them, in Section 7.

Limitations

Two kinds of caveat apply to the claims above, and they are different in nature. The first kind is a genuine threat to the validity of a result, and we hedge it accordingly. The second is scope we set deliberately, which bounds where a claim applies without weakening it, and which we take up as future work in Section 8. For each threat we also name the specific experiment that would resolve it, so that the open questions read as a concrete experimental map rather than as unspecified doubt.

Scale and fit (threat). Both SSSM and the baseline are trained at a 14.4B-token budget at which neither is fully fit, and their zero-shot benchmark numbers sit near chance. We therefore read those numbers as indicative of broad trends rather than as a mature evaluation. This bounds the behavioral comparison most directly. It also reaches the mechanistic probe more than we would like, since the gate behavior of an underfit model need not match its behavior at convergence, and we state the probe as an observation at this budget rather than as a converged property. The experiment that would resolve it is to train both models closer to convergence and re-run the probe, checking that the Δt separation persists. Separately, the results are from a single training run per condition, so run-to-run variance is not characterized, and a stronger version would repeat each condition over several seeds, at least for the probe.

The pretrained sentence encoder confounds the granularity claim (threat). SSSM reads and writes through a SONAR encoder and decoder pretrained on large multilingual data, whereas the token baseline learns its representations from scratch. Part of SSSM's sharper selectivity may therefore come from the quality of that pretrained space rather than from sentence granularity. As Section 6 notes, the contrast is asymmetric, since the token baseline never uses the encoder and so its failure to concentrate on a multi-token unit is attributable to granularity alone, but the SSSM side is exposed to this confound. The experiment that would isolate granularity is a control in which the sentence encoder is weakened or randomly initialized, or in which the token baseline is given an equivalent pretrained front end, so that the two sides are matched in representation quality.

The decoder confounds the text-level comparison (threat). The SONAR decoder is fine-tuned together with the backbone (Section 3.2), so the text-level gains in Section 5.2 reflect the combined effect of the backbone and the fine-tuned decoder rather than sentence-level granularity in isolation. The mechanistic probe is the part of the evidence least exposed to this confound, because it reads the gate statistic before any decoding takes place. The experiment that would resolve it is a frozen-decoder ablation, in which the SONAR decoder is held fixed so that any remaining text-level gain is attributable to the backbone alone.

Selectivity is not yet linked to capability (threat). The probe reports where the gate writes most strongly, not whether SSSM solves the variable-tracking task. We measured that accuracy, and both models score zero at this budget. We read this as a floor effect rather than as counter-evidence to the probe. Selecting a sentence is the write step. Answering the query also needs a readout step that neither underfit model has learned, and the two belong to different stages of the computation. The experiment that would connect them is a readout probe on the hidden state, testing whether the value carried by the selected chain sentences is recoverable downstream, reported alongside variable-tracking accuracy.

Scope of the mechanistic claim. The gate-selectivity result is established on one task family, RULER variable-tracking, under teacher forcing, and we state it as an observation at that scope. The chain sentences are also distributionally distinct from the natural-language distractors, since a `VAR _ = _` line is unlike ordinary prose, so the complete separation we observe may partly reflect how separable that pattern is in SONAR space rather than a learned ranking of importance. A stronger test would use a task whose important and unimportant sentences are closer in distribution. Whether the same selectivity holds there, on other importance-structured inputs, and under autoregressive decoding is a question of generality, and we leave it to future work. The decoding-path control likewise shows that the predicted vector, while well below the no-signal bound, stays above the round-trip upper bound, so it is informative without being a faithful reconstruction.

Deliberately deferred scope. Two further boundaries are choices rather than threats. We adopt the

joint training regime (Section 3.4) on qualitative grounds and on observations made during training, without the matched ablation of the three regimes that would settle the choice quantitatively. And we define a state-action packing format, detailed in the Auxiliary Objective and Packing Format appendix, but do not train or evaluate with it, so adapting a pretrained SSSM to structured downstream data is left open. Both are taken up as future work in Section 8.

Conclusion

We asked whether the token is the right granularity for Mamba's selective scan, or whether a more complete semantic unit would suit the gating mechanism better. To make the question testable we built SentenceSSM, which keeps the Mamba2 backbone unchanged and replaces only its input unit, from token embeddings to whole-sentence SONAR vectors, and we compared it against a token-level Mamba2 at a matched parameter and token budget.

Our evidence is consistent across the two levels we examined. Behaviorally, sentence granularity does not cost general language ability and coincides with higher decoded-text quality, although this comparison is confounded by the jointly fine-tuned decoder. Mechanistically, the gate under sentence-level input concentrates its largest Δt on the a priori important sentences, with a mean rank of about 5 out of roughly 710 and a mean z-score of $+7.11$, while the same mechanism on a token stream does not, and a decoding-path control shows that the predicted vector carries genuine sentence information rather than only a decoder prior. We read this as a mechanistic observation consistent with the granularity-mismatch hypothesis, not as proof of it. The contrast also reflects SSSM's pretrained sentence encoder, and only part of it, the token model's failure to concentrate on a multi-token unit, is attributable to granularity alone.

Several directions would extend this, and the most important separate granularity from the pretrained encoder. A control that weakens or randomly initializes the sentence encoder, or that gives the token baseline an equivalent pretrained front end, would test whether the selectivity gap survives when the two sides are matched in representation quality. A readout probe on the hidden state would test whether the selected sentences translate into higher variable-tracking accuracy, connecting selectivity to capability. Beyond these, a matched ablation of the training regimes would put the choice of joint training on firmer ground, extending the probe beyond a single task family and to the autoregressive setting would test how general the effect is, and repeating the comparison at a larger budget, where both models are better fit, would tell whether the pattern persists at

convergence.

Taken as a whole, this work delivers a new architecture, a mechanistic probe for it, and one clean differential observation. At the sentence granularity the gate concentrates selection on importance-structured input in a way the token stream does not elicit at matched scale. We present this as a well-posed and falsifiable starting point rather than a settled result. It frames the view that the token may be too fine a unit for selective state-space models as a direction worth pursuing, and it names the experiments that would test it.

References

[1] Albert Gu and Tri Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. In *Conference on Language Modeling (COLM)*, 2024. arXiv:2312.00752.

[2] Tri Dao and Albert Gu. Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality. In *International Conference on Machine Learning (ICML)*, 2024. arXiv:2405.21060.

[3] Paul-Ambroise Duquenne, Holger Schwenk, and Benoît Sagot. SONAR: Sentence-Level Multimodal and Language-Agnostic Representations. 2023. arXiv:2308.11466.

[4] 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. In *International Conference on Machine Learning (ICML)*, 2024. arXiv:2402.04248.

[5] Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norick, Vijay Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, Jan Kautz, Mohammad Shoeybi, and Bryan Catanzaro. An Empirical Study of Mamba-based Language Models. 2024. arXiv:2406.07887.

[6] Samy Jelassi, David Brandfonbrener, Sham M. Kakade, and Eran Malach. Repeat After Me: Transformers are Better than State Space Models at Copying. In *International Conference on Machine Learning (ICML)*, 2024. arXiv:2402.01032.

[7] 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. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2501.00658.

[8] Zhifan Ye, Kejing Xia, Yuhong Fu, Xinyu Dong, Jihoon Hong, Xiangchi Yuan, Shizhe Diao, Jan Kautz, Ali Hatamizadeh, and Yingyan Lin. LongMamba: Enhancing Mamba's Long-

Context Capabilities via Training-Free Receptive Field Enlargement. In *International Conference on Learning Representations (ICLR)*, 2025.

[9] Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated Delta Networks: Improving Mamba2 with Delta Rule. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2412.06464.

[10] Sheridan Feucht, David Atkinson, Byron C. Wallace, and David Bau. Token Erasure as a Footprint of Implicit Vocabulary Items in LLMs. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024. arXiv:2406.20086.

[11] Guy Kaplan, Matanel Oren, Yuval Reif, and Roy Schwartz. From Tokens to Words: On the Inner Lexicon of LLMs. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.05864.

[12] Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. ByT5: Towards a Token-Free Future with Pre-trained Byte-to-Byte Models. *Transactions of the Association for Computational Linguistics (TACL)*, 10:291–306, 2022. arXiv:2105.13626.

[13] Artidoro Pagnoni, Ram Pasunuru, Pedro Rodriguez, John Nguyen, Benjamin Muller, Margaret Li, Chunting Zhou, Lili Yu, Jason Weston, Luke Zettlemoyer, Gargi Ghosh, Mike Lewis, Ari Holtzman, and Srinivasan Iyer. Byte Latent Transformer: Patches Scale Better Than Tokens. 2024. arXiv:2412.09871.

[14] Junxiong Wang, Tushaar Gangavarapu, Jing Nathan Yan, and Alexander M. Rush. MambaByte: Token-free Selective State Space Model. In *Conference on Language Modeling (COLM)*, 2024. arXiv:2401.13660.

[15] Nasim Borazjanizadeh and James L. McClelland. Modeling Language as a Sequence of Thoughts. 2025. arXiv:2512.25026.

[16] Loïc Barrault, Paul-Ambroise Duquenne, Maha Elbayad, Artyom Kozhevnikov, Belen Alastruey, Pierre Andrews, Mariano Coria, Guillaume Couairon, Marta R. Costa-jussà, David Dale, Hady Elsahar, Kevin Heffernan, João Maria Janeiro, Tuan Tran, Christophe Ropers, Eduardo Sánchez, Robin San Roman, Alexandre Mourachko, Safiyyah Saleem, and Holger Schwenk. Large Concept Models: Language Modeling in a Sentence Representation Space. 2024. arXiv:2412.08821.

[17] Nikita Dragunov, Temurbek Rahmatullaev, Elizaveta Goncharova, Andrey Kuznetsov, and Anton Razzhigaev. SONAR-LLM: Autoregressive Transformer that Thinks in Sentence Embeddings and Speaks in Tokens. 2025. arXiv:2508.05305.

[18] Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural Network Acceptability Judgments. *Transactions of the Association for Computational Linguistics (TACL)*, 7:625–641, 2019. arXiv:1805.12471.

[19] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. 2018. arXiv:1803.05457.

[20] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a Machine Really Finish Your Sentence? In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019. arXiv:1905.07830.

[21] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about Physical Commonsense in Natural Language. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020. arXiv:1911.11641.

[22] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An Adversarial Winograd Schema Challenge at Scale. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020. arXiv:1907.10641.

[23] Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A Corpus and Cloze Evaluation for Deeper Understanding of Commonsense Stories. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2016. arXiv:1604.01696.

[24] Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the Real Context Size of Your Long-Context Language Models? In *Conference on Language Modeling (COLM)*, 2024. arXiv:2404.06654.

Appendices

Implementation and Reproducibility

The model referred to as SSSM is implemented as `ConceptMamba`, a Mamba2 mixer stack extended to accept embedding inputs, using the official `mamba-ssm==2.2.4` selective-scan kernel. The training loop wraps PyTorch Accelerate with FSDP and records checkpoints by cumulative token count rather than by step count.

Training Hyperparameters and Regimes

The 780M-parameter SSSM is trained on a single node with four A100 GPUs. The salient hyperparameters are:

- Per-GPU batch size: 8 trajectories, gradient accumulation: 8, for an effective batch of 256 trajectories per update.
- Maximum trajectory length: 64 sentences.
- Learning rate schedule: warmup-stable-decay (WSD) with 300 warmup steps over the 6,588-step pretrain. The decay phase is reported as a separate continuation of the same checkpoint, run on a 10% data resample.
- Optimizer: AdamW, mixed precision bfloat16, gradient clipping at 1.0.
- Reported milestones: results are recorded at 1B, 5B, 10B, and 14.4B cumulative tokens.

SSSM couples a trainable SSM backbone with a SONAR decoder, and the two freeze switches, whether the backbone is frozen and whether the decoder is trained, yield the three regimes in Table 7. The primary checkpoint uses the joint regime (b), in which the backbone and the decoder are trained at the same time, each with its own learning rate, so that the decoder can adapt to

the backbone's output distribution in SONAR space. Regime (a) cannot do this, because there the decoder is a fixed reconstructor that must consume whatever the backbone emits. Regime (c) achieves it only partially, since freezing the backbone before training the decoder forgoes the chance to co-adapt the two. The choice of (b) rests on this qualitative analysis together with preliminary observations made during training, not on a controlled comparison, and the matched ablation of the three regimes that would support it is left to future work.

Regime	`freeze_ssm`	`train_decoder`	**Meaning**
(a) train SSM, freeze decoder	F	F	decoder is a fixed reconstructor
(b) joint *(primary)*	F	T	backbone and decoder trained together (two optimizers, independent learning rates)
(c) two-stage	T	T	backbone trained first, then frozen while the decoder is trained

Table 7: The three training regimes induced by the two freeze switches. The primary checkpoint uses the joint regime (b).

Auxiliary Objective and Packing Format

MSE objective. The MSE control variant of Section 3.2 trains an otherwise-identical model on a mean-squared error between the predicted next-sentence embedding $\hat{x}_{t+1}$ and the ground-truth next-sentence embedding x_{t+1}. For a packed trajectory of length T,

$$\mathscr{L}_{\text{MSE}} = \frac{1}{T-1} \sum_{t=1}^{T-1} \left\| \hat{x}_{t+1} - x_{t+1} \right\|_2^2.$$

It optimizes only the geometric placement of the predicted vector in SONAR space, and because embedding-space proximity is a loose proxy for decoded-text quality, the primary checkpoint uses the cross-entropy objective of Section 3.2 instead.

State-action packing format. Besides the plain trajectory stream used for pretraining (Section 3.2), we define a richer packing format for data with an explicit state and action structure, in which state and action sentences are packed together with dedicated boundary markers. We define this format but do not train or evaluate with it in this work, and adapting a pretrained SSSM to structured downstream data through it is left as future work.

Dataset Splits

ROC-Stories [23] has train, validation, and test splits of 62.9k, 7.7k, and 7.9k trajectories. The SSSM run covers 15,177 train-split trajectories and the baseline run covers the full 7,909-trajectory test split. These two runs are intersected by ground truth to the 1,477-trajectory matched subset on which all cross-architecture generation metrics are reported (Section 5.2).

Supporting Figures

These single-sample heatmaps illustrate the aggregate gate-selectivity result of Section 5.4 on one trajectory.

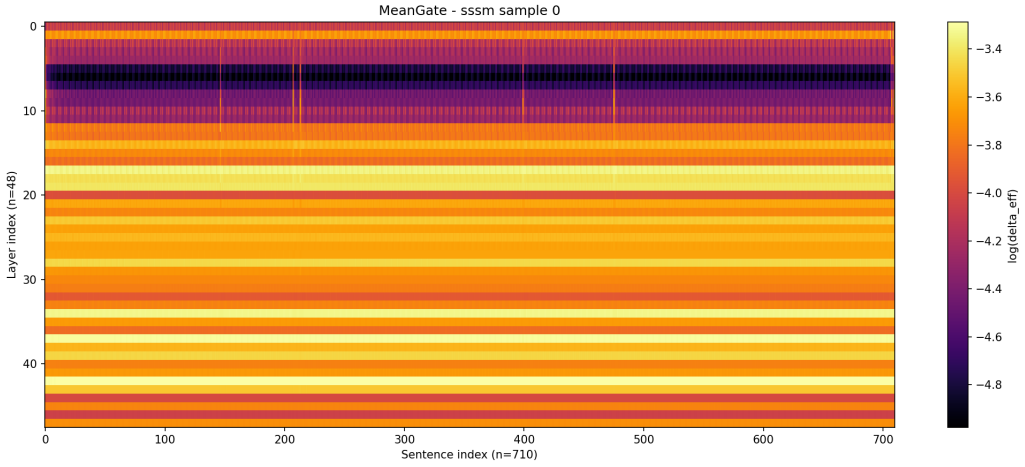

Figure 2: Per-layer mean gate Δt for SSSM on one RULER variable-tracking sample (48 layers by 710 sentences, log scale, clipped at the 1st and 99th percentiles). The bright band marks the chain sentences, which sit at the top of the Δt distribution, every one within the top 12 of about 710.

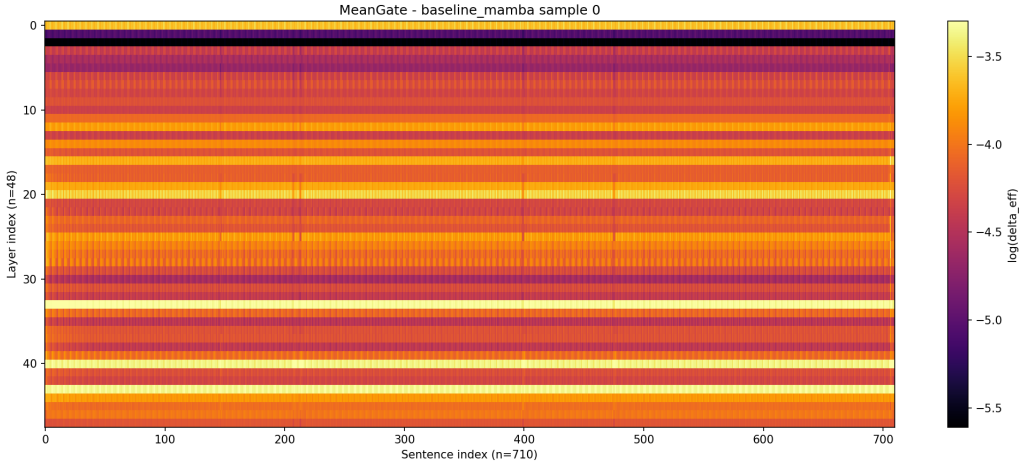

Figure 3: Per-layer mean gate Δt for the token-level baseline on the same sample as Figure 2, on the same scale. The chain sentences are not singled out.