Selectivity Makes State-Space Models Universal Sequence Approximators

Yoshihiro Maruyama
Department of Mathematical and Information Sciences
Kyoto University
maruyama@i.h.kyoto-u.ac.jp

Abstract

Selective state-space models such as Mamba replace fixed linear state transitions by input-dependent transitions, yielding recurrent sequence models that retain linear-time inference at fixed model size while adapting their memory dynamics to the current token. We show that even a simple form of this selective memory update has universal approximation power. Specifically, for every compact input domain, finite horizon, and continuous causal sequence-to-sequence map, a clocked diagonal selective state-space scan with strictly contractive gates and a linear-in-state readout uniformly approximates the target map. The proof is constructive: input-dependent diagonal propagation generates tensor-product partition features over prefixes, and the linear-in-state readout combines these features with sampled target values. Thus the high-order interactions required for universality are produced inside the selective scan itself, rather than by attention, nonlinear hidden-state dynamics, or a universal multilayer perceptron applied after a sequence encoder. These results identify selectivity as a mathematically sufficient mechanism for expressive linear-time state-space sequence modeling. Selectivity lets the model decide, token by token, which parts of the past should be preserved, faded, or combined with the present. The universal approximation theorem shows that this adaptive use of memory is sufficient to reproduce any well-behaved causal sequence rule.

1 Introduction

Sequence modeling has involved a tradeoff between how flexibly a model can use its context and how efficiently it can process long streams. Recurrent neural networks process tokens causally with a hidden state that is updated one step at a time (Rumelhart et al., 1986; Elman, 1990). Gated recurrent architectures such as LSTMs and GRU encoder-decoders improved the ability of this state to preserve and update information over longer ranges (Hochreiter and Schmidhuber, 1997; Cho et al., 2014). This recurrent viewpoint is attractive because inference is naturally linear in sequence length, but the model must compress all relevant past information into a finite state whose update rule determines which interactions across time can be represented.

Attention shifted the emphasis from compression into a single recurrent state to content-dependent access over previously seen tokens. Alignment mechanisms and Transformers made it possible for representations at one position to depend directly on representations at other positions (Bahdanau et al., 2015; Vaswani et al., 2017). This flexibility has been central to modern sequence modeling, but full attention scales quadratically with sequence length. As context lengths grow, this cost has motivated efficient-attention, linear-attention, and retention-style alternatives that try to keep some of the content-dependent behavior of attention while improving computational scaling (Tay et al., 2022; Katharopoulos et al., 2020; Choromanski et al., 2021; Sun et al., 2023).

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

State-space models (SSMs) provide another route through this tradeoff. In an SSM, a latent state is propagated by a dynamical system and then read out to produce sequence outputs. Modern structured SSMs exploit this form to obtain recurrent or convolutional implementations, making them well suited to long sequences (Gu et al., 2020, 2021, 2022a). Subsequent work showed that diagonal and simplified state-space parameterizations can be highly effective in practice (Gu et al., 2022b; Gupta et al., 2022; Smith et al., 2023), and SSM-inspired architectures have become competitive in language and long-context modeling (Fu et al., 2023; Mehta et al., 2023; Orvieto et al., 2023; Peng et al., 2023; Poli et al., 2023). The common appeal is that these models retain a scan-like computation: they can process long sequences without explicitly forming all pairwise token interactions.

The same structure that makes classical SSMs efficient also raises an expressivity question. A fixed-transition SSM carries information forward through a transition rule that does not depend on the current token. Such a model can accumulate transformed inputs and store long-range summaries, but its memory dynamics are not directly conditioned on whether the current token should preserve, erase, or combine particular features from the past. In contrast, Mamba-style selective state-space models make key parts of the state-space update input dependent, including the discretization, input, and output maps (Gu and Dao, 2024; Dao and Gu, 2024). At the scan level, the state update remains affine in the previous state once the current token is fixed, with coefficients selected by the token itself.

This distinction is the starting point of the paper. Selectivity could be viewed merely as a useful parameterization for training large sequence models, or it could represent a genuine change in what the scan can express. The latter question is not answered by empirical performance alone. It is also not answered by classical universal approximation theorems for multilayer perceptrons or recurrent neural networks (Cybenko, 1989; Hornik, 1991; Funahashi and Nakamura, 1993; Schäfer and Zimmermann, 2007), or by more recent universality results for attention, temporal convolutional, and stacked SSM/recurrent sequence models (Hanson and Raginsky, 2019; Yun et al., 2020a,b; Alberti et al., 2023; Wang and Xue, 2023; Kajitsuka and Sato, 2024; Petrov et al., 2024a,b; Cheng et al., 2025). Those results apply to broad nonlinear architectures or to different sequence mechanisms; they do not clarify whether the selective SSM recurrence itself, with a diagonal transition and a linear readout, is capable of generating the cross-token interactions needed for arbitrary causal sequence maps.

We answer this question by proving a universal approximation theorem for a diagonal selective SSM. On any compact input domain and any fixed finite horizon, the model can uniformly approximate every continuous causal sequence-to-sequence map. The theorem is formulated for the following type of architecture: the hidden-state update is diagonal and affine in the previous state, the gates are strictly contractive, and the readout is linear in the state. The result therefore does not rely on attention, nonlinear hidden-state activations, or a universal multilayer perceptron placed after a sequence encoder, proving that token-dependent selection in the scan is already sufficient for finite-horizon causal universality.

The constructive proof gives a concrete reason this is possible. A selective diagonal gate can multiply a stored prefix feature by a function of the current token. Repeating this operation over time lets the scan build tensor-product features over prefixes: each state coordinate can be interpreted as following one path through a soft partition of the input space. Once these prefix features are present in the state, a linear readout can combine them with sampled target values to approximate any continuous causal map on the compact domain. This mechanism is unavailable to fixed-transition additive SSMs, which can store sums of tokenwise features but cannot generally create mixed interactions such as products of two earlier tokens inside the recurrent state.

The contributions of this paper can be summarized as follows:

- For clocked diagonal selective SSMs capturing the input-dependent transition of S6, Mamba's selective state-space layer, we prove a constructive universal approximation theorem for continuous causal maps on compact finite-horizon domains.
- We rigorously prove a separation from a broad fixed-transition additive SSM class, showing that selectivity supplies expressive power not present in fixed linear memory alone.
- We show that an autonomous selective SSM layer generates the clock internally, and that finite neural selector networks achieve the approximation guarantee with strictly contractive exponential gates.

These results identify selectivity as a mathematically sufficient mechanism for expressive linear-time state-space sequence modeling. Appendices A and B further develop these results, detailing the correspondence with Mamba/S6 and showing how residual depth and structured coefficient factorizations organize and compress the features generated by the scan. The compression result gives a concrete design principle: accurate factorizations of sampled target tensors with few product terms and nonzero nonnegative input-mode factors yield smaller recurrent realizations with controlled approximation error.

The remainder of the paper is organized as follows. Section 2 defines the selective scan model; Section 3 proves the universal approximation theorem and discusses its resource bounds; Section 4 establishes the fixed-transition separation, including clock-dependent input and readout maps; Sections 5 and 6 discuss autonomous clocks, neural realization, and connections to existing universal approximation results. The appendices give the architectural correspondence and the proofs of the neural, residual, and factored realizations.

2 Mamba-style selective state-space scans

2.1 Causal maps and the uniform approximation problem

Fix a finite horizon $T \in \mathbb{N}$, an input dimension d, an output dimension m, and a compact input set $X \subset \mathbb{R}^d$. A sequence-to-sequence map

$$F : X^T \to (\mathbb{R}^m)^T, \qquad F(x_{1:T}) = (F_1(x_1), F_2(x_{1:2}), \dots, F_T(x_{1:T})) \tag{1}$$

is called *causal* if each coordinate map $F_t : X^t \to \mathbb{R}^m$ depends only on the prefix $x_{1:t}$. Throughout, continuity of F means that every F_t is continuous on the compact product domain X^t.

We measure approximation error in the uniform causal norm

$$\left\| F - \widehat{F} \right\|_{\infty,T} := \sup_{x_{1:T} \in X^T} \max_{1 \le t \le T} \left\| F_t(x_{1:t}) - \widehat{F}_t(x_{1:t}) \right\|_2 . \tag{2}$$

The finite-horizon and compact-domain assumptions are standard for uniform approximation theorems.

2.2 Clocked diagonal selective SSMs

Let $\mathcal{E}_T = \{e_1, \dots, e_T\} \subset \mathbb{R}^T$ denote the standard one-hot clock codes. The clock is used to state a one-layer theorem clearly; it can be generated internally by a preliminary SSM layer, as discussed later in the paper.

Definition 2.1 (Clocked diagonal selective SSM)**.** A clocked diagonal selective state-space model of width N is specified by continuous selector functions

$$\lambda : X \times \mathcal{E}_T \to (0,1)^N, \qquad b : X \times \mathcal{E}_T \to \mathbb{R}^N, \qquad C : X \times \mathcal{E}_T \to \mathbb{R}^{m \times N}, \tag{3}$$

and computes, for $u_t = (x_t, e_t)$,

$$h_0 = 0, \qquad h_t = \lambda(u_t) \odot h_{t-1} + b(u_t), \qquad \widehat{y}_t = C(u_t) h_t. \tag{4}$$

Here $\odot$ denotes elementwise multiplication. The gates are *strictly contractive* because every coordinate of $\lambda(u)$ lies in $(0,1)$. The readout matrix $C(u_t)$ may depend on the current input-clock pair, but for each fixed u_t the map $h_t \mapsto C(u_t)h_t$ is linear in the state.

Since $\mathcal{E}_T$ is finite, continuity in Definition 2.1 simply means that the time-slice functions $x \mapsto \lambda(x, e_t)$, $x \mapsto b(x, e_t)$, and $x \mapsto C(x, e_t)$ are continuous on X. The state update is affine in the previous state, with a diagonal linear part, and the readout is linear in the hidden state. An input-dependent readout can create interactions between the current token and earlier tokens. For each fixed current token, the output of a fixed-transition additive SSM is additive across earlier tokens; this structure underlies the separation in Proposition 4.2. In our construction, the input-dependent diagonal transition builds tensor-product features across tokens inside the recurrent state. A readout that depends only on the clock combines these features to approximate the target causal map.

Section 5 replaces the external one-hot clock with a scalar recurrent clock generated by a preliminary layer. The recurrence $c_t = \rho c_{t-1} + 1$, with $c_0 = 0$ and $0 < \rho < 1$, produces distinct values at the first T scan steps.

2.3 Mamba/S6 parameterization

The recurrence (4) is the diagonal selective-scan abstraction of the input-dependent state-space update used in Mamba/S6 (Gu and Dao, 2024). In a continuous-time diagonal SSM with negative diagonal generator $A = \mathrm{diag}(A_1, \dots, A_N)$, $A_i < 0$, and input-dependent discretization step $\Delta_i(u_t) > 0$, the discrete transition has coordinates

$$\bar{A}_i(u_t) = \exp(A_i \Delta_i(u_t)). \tag{5}$$

Thus every coordinate gate lies in $(0, 1)$. Conversely, every strictly contractive gate in Definition 2.1 can be written in this form by choosing, for example,

$$A_i = -1, \qquad \Delta_i(u) = -\log \lambda_i(u) > 0. \tag{6}$$

The additive term $b(u_t)$ can be viewed as the discretized input contribution after adjoining a constant input channel. For example, exact zero-order-hold discretization with generator -1 gives the scalar coordinate form

$$h_t^i = e^{-\Delta_i(u_t)} h_{t-1}^i + \big(1 - e^{-\Delta_i(u_t)}\big) B_i(u_t), \tag{7}$$

and one obtains (4) by setting

$$B_i(u) = \frac{b_i(u)}{1 - \lambda_i(u)}. \tag{8}$$

This function is continuous on the compact clocked input domain whenever $0 < \lambda_i(u) < 1$.

The theorem uses the generalized scan of Definition 2.1 with continuous selectors. Its construction is finitely parameterized by Gaussian centers, their scale, and sampled target coefficients; its neural realization uses finite selector networks from a uniformly dense family, with network size allowed to grow. Stock Mamba/S6 additionally imposes channel-sharing and projection constraints. Appendix A details these constraints, the correspondence at the scan interface, the surrounding mixer, and both input-discretization conventions.

3 Universal approximation by selective tensor partitions

3.1 Main theorem

The main theorem states that the model in Definition 2.1 is dense in the space of continuous causal maps on a fixed compact horizon. The construction uses a finite net of the input domain and creates one state coordinate for each possible prefix of net indices. A finite r-net of X means points $\xi_1, \dots, \xi_M \in X$ such that every $x \in X$ satisfies $\min_j \|x - \xi_j\|_2 \le r$.

Theorem 3.1 (Selective tensor-partition universal approximation)**.** *Let $X \subset \mathbb{R}^d$ be compact, let $T \in \mathbb{N}$, and let*

$$F : X^T \to (\mathbb{R}^m)^T, \qquad F(x_{1:T}) = (F_1(x_1), \dots, F_T(x_{1:T})) \tag{9}$$

be continuous and causal. For every $\varepsilon > 0$, there exists a clocked diagonal selective SSM of the form (4)*, with strictly contractive gates and a linear-in-state readout whose matrix depends only on the clock, such that*

$$\sup_{x_{1:T} \in X^T} \max_{1 \le t \le T} \|\widehat{y}_t(x_{1:t}) - F_t(x_{1:t})\|_2 < \varepsilon. \tag{10}$$

Moreover, the construction uses a finite r-net of X for a sufficiently small $r > 0$; if M is the size of that net, the width is

$$N = \sum_{\ell=1}^{T} M^\ell. \tag{11}$$

Proof. First, we approximate each target output by a weighted average of its values at representative prefixes. We then construct a selective scan that generates the product weights appearing in this average.

If $X = \varnothing$, the assertion is vacuous, so assume X is nonempty. Let $\varepsilon > 0$ be fixed and set

$$R = \max_{1 \le t \le T} \sup_{z \in X^t} \|F_t(z)\|_2, \qquad B = \max\{1, R\}. \tag{12}$$

The number R is finite because each F_t is continuous on a compact domain. By uniform continuity of the finitely many maps F_t, there exists $\delta > 0$ such that, for every $t = 1, \dots, T$ and all $x_{1:t}, z_{1:t} \in X^t$,

$$\max_{1 \le s \le t} \|x_s - z_s\|_2 < \delta \quad \Longrightarrow \quad \|F_t(x_{1:t}) - F_t(z_{1:t})\|_2 < \frac{\varepsilon}{2}. \tag{13}$$

(i) Tensor partition approximation. Choose a finite $\delta/2$-net $\xi_1, \dots, \xi_M \in X$. If necessary, duplicate one net point so that $M \ge 2$. For $\sigma > 0$, define the normalized Gaussian partition

$$\phi_j(x) = \frac{\exp(-\|x - \xi_j\|_2^2 / \sigma^2)}{\sum_{k=1}^{M} \exp(-\|x - \xi_k\|_2^2 / \sigma^2)}, \tag{14}$$

so every ϕ_j is continuous and strictly between 0 and 1. These functions satisfy

$$\phi_j : X \to (0, 1), \qquad \sum_{j=1}^{M} \phi_j(x) = 1. \tag{15}$$

We choose σ small enough to make the partition uniformly concentrated near the net points. For any $x \in X$, let $a(x)$ satisfy $\|x - \xi_{a(x)}\|_2 \le \delta/2$. If $\|x - \xi_j\|_2 \ge \delta$, then

$$\phi_j(x) \le \exp\left(-\frac{\delta^2 - (\delta/2)^2}{\sigma^2}\right). \tag{16}$$

Hence

$$\sum_{\{j : \|x - \xi_j\|_2 \ge \delta\}} \phi_j(x) \le M \exp\left(-\frac{3\delta^2}{4\sigma^2}\right). \tag{17}$$

Taking σ sufficiently small gives the uniform bound

$$\sum_{\{j : \|x - \xi_j\|_2 \ge \delta\}} \phi_j(x) < \frac{\varepsilon}{4BT} \qquad \text{for every } x \in X. \tag{18}$$

For each time t, define the tensor-partition approximant

$$Q_t(x_{1:t}) = \sum_{j_1, \dots, j_t = 1}^{M} F_t(\xi_{j_1}, \dots, \xi_{j_t}) \prod_{s=1}^{t} \phi_{j_s}(x_s). \tag{19}$$

The weights in (19) are nonnegative and sum to one. A tuple $(j_1, \dots, j_t)$ is good for $x_{1:t}$ if $\|x_s - \xi_{j_s}\|_2 < \delta$ for every $s \le t$, and bad otherwise. By the union bound and (18), the total weight of bad tuples is less than $\varepsilon/(4B)$. For good tuples, (13) bounds the summand error by $\varepsilon/2$. For bad tuples, the summand error is at most $2R$. Therefore, for every $t = 1, \dots, T$,

$$\sup_{x_{1:t} \in X^t} \|Q_t(x_{1:t}) - F_t(x_{1:t})\|_2 < \varepsilon, \tag{20}$$

because $\varepsilon/2 + 2R\varepsilon/(4B) \le \varepsilon$.

(ii) Scan realization. It remains to show that the features in (19) can be generated by a diagonal selective SSM. Index state coordinates by tuples

$$q = (\ell, j_1, \dots, j_\ell), \qquad 1 \le \ell \le T, \qquad j_s \in \{1, \dots, M\}. \tag{21}$$

Here ℓ is the time at which coordinate q is read, and r denotes the current scan step. There are exactly $N = \sum_{\ell=1}^{T} M^\ell$ such coordinates. For the coordinate $q = (\ell, j_1, \dots, j_\ell)$, define the additive selector

$$b_q(x, e_r) = \begin{cases} \phi_{j_1}(x), & r = 1, \\ 0, & r \ne 1, \end{cases} \tag{22}$$

and define the diagonal gate

$$\lambda_q(x, e_r) = \begin{cases} \phi_{j_r}(x), & 2 \le r \le \ell, \\ 1/2, & \text{otherwise.} \end{cases} \tag{23}$$

These selectors are continuous on $X \times \mathcal{E}_T$ and strictly contractive because each ϕ_j lies in $(0, 1)$.

We claim that for every coordinate $q = (\ell, j_1, \ldots, j_\ell)$ and every $1 \le t \le \ell$,

$$h_t^q = \prod_{s=1}^{t} \phi_{j_s}(x_s). \tag{24}$$

Indeed, $h_0^q = 0$, and at time $t = 1$,

$$h_1^q = \lambda_q(x_1, e_1) h_0^q + b_q(x_1, e_1) = \phi_{j_1}(x_1). \tag{25}$$

If (24) holds at time $t - 1$ for some $2 \le t \le \ell$, then by (23),

$$h_t^q = \phi_{j_t}(x_t) h_{t-1}^q = \prod_{s=1}^{t} \phi_{j_s}(x_s). \tag{26}$$

Thus, up to time ℓ, the selective diagonal transition implements exactly the corresponding prefix of the tensor product that appears in (19). The coordinate $q = (\ell, j_1, \ldots, j_\ell)$ is read only at clock time ℓ; its values at earlier or later times are ignored by the readout below.

Finally define the readout on each clock slice to use only coordinates whose length matches the clock. In column notation, for $r = 1, \ldots, T$ set

$$C_{:,q}(x, e_r) = \begin{cases} F_r(\xi_{j_1}, \ldots, \xi_{j_r}), & q = (r, j_1, \ldots, j_r), \\ 0_{\mathbb{R}^m}, & \text{otherwise.} \end{cases} \tag{27}$$

Then (24) gives

$$\widehat{y}_t = \sum_{j_1, \ldots, j_t = 1}^{M} F_t(\xi_{j_1}, \ldots, \xi_{j_t}) \prod_{s=1}^{t} \phi_{j_s}(x_s) = Q_t(x_{1:t}). \tag{28}$$

Combining (20) and (28) proves Theorem 3.1. □

The proof shows more than density. It gives an explicit feature family generated by the scan. Intuitively, each state coordinate follows one path through a soft partition tree over the prefix, and the selective gate appends the current token's soft bin weight to that path.

3.2 The expressivity mechanism

Equation (24) is the central mechanism. A fixed additive SSM can store sums of transformed tokens, but it cannot multiply a stored prefix feature by a new current-token selector inside its recurrent state. In contrast, the selective recurrence

$$h_t^q = \lambda_q(x_t, e_t) h_{t-1}^q + b_q(x_t, e_t) \tag{29}$$

uses the input-dependent gate itself as a multiplicative operator. This converts local partition functions $\phi_j(x_t)$ into prefix tensor features $\prod_{s \le t} \phi_{j_s}(x_s)$. The theorem therefore identifies selective multiplication, rather than attention or hidden-state nonlinearities, as a sufficient mechanism for finite-horizon causal universality.

Strict contraction is compatible with this expressive mechanism. For a fixed input and clock, the state update contracts differences between hidden states, while its token-dependent coefficients create interactions across input positions. In the construction, the local partition weights describe soft matches to representative inputs, and their products describe joint matches to representative prefixes. At each prefix length these products sum to one, so the readout forms a weighted average of sampled target values. Universal finite-horizon causal approximation is thus achieved within a family of strictly contractive scans.

3.3 Resource bounds and structured realizations

The width bound (11) gives $N = \Theta(M^T)$ for fixed $M \ge 2$, providing an explicit resource characterization of the tensor-partition construction. The partition size is chosen according to the target, horizon, and approximation accuracy. A width-N scan performs $O(TN)$ recurrent-update operations,

with selector and readout evaluation counted separately, so its scan cost is linear in sequence length at fixed model size.

The same approximant admits an exact depth-width tradeoff: for any integer $W \geq 1$, its coordinates can be distributed among $\lceil N/W \rceil$ residual selective blocks, each of recurrent width at most W, while preserving the total feature-state count. Each block receives the original token and clock, and their readouts accumulate in a pointwise residual stream.

The sampled coefficient tensor is a table of target outputs at representative input histories. A factorization expresses this table as a sum of products of factors associated with individual time positions. Averaging each factor with the local partition weights gives a function of the corresponding token, so one recurrent coordinate can build each resulting product by successive multiplication. When an accurate factorization uses nonzero nonnegative mode factors, normalization makes these token functions valid strictly contractive gates, with the scale absorbed into the readout. A short factorization therefore saves state: R_t product terms replace the M^t coordinates for prefixes of length t, giving total recurrent width $\sum_t R_t$. A maximum coefficient error δ_t contributes at most δ_t to the uniform output error at time t. Appendix B proves both constructions and explains how their error bounds combine with finite neural selector approximation.

4 Selectivity creates interactions unavailable to fixed SSMs

The proof of Theorem 3.1 uses one operation unavailable to fixed-transition additive SSMs as a state update: multiplication of a stored prefix feature by a current-token selector. This section formalizes the distinction by comparing Definition 2.1 with the following broad fixed-transition additive class.

Definition 4.1 (Fixed-transition additive SSM)**.** A finite-dimensional fixed-transition additive SSM on scalar inputs and scalar outputs is specified by a dimension $n < \infty$, a fixed matrix $A \in \mathbb{R}^{n \times n}$, and continuous functions

$$\beta : [-1, 1] \to \mathbb{R}^n, \qquad \gamma : [-1, 1] \to \mathbb{R}^n, \qquad \kappa : [-1, 1] \to \mathbb{R}. \tag{30}$$

It computes

$$z_0 = 0, \qquad z_t = Az_{t-1} + \beta(x_t), \qquad y_t = \gamma(x_t)^\top z_t + \kappa(x_t), \tag{31}$$

for $t \geq 1$. The transition matrix is input independent, while the additive input map and the readout coefficients may depend arbitrarily and continuously on the current token. At each fixed x_t, the readout is linear in z_t up to the scalar offset $\kappa(x_t)$.

The current-token-dependent readout allows this class to form interactions between the present input and stored information. To examine interactions between earlier tokens, think of x_1 and x_2 as signed observations, with $x_1 x_2$ measuring their agreement. Requiring this output at time three, with x_3 held fixed, tests whether the recurrent state has formed a joint feature of the earlier observations that its linear-in-state readout can recover.

Proposition 4.2 (A fixed-transition obstruction)**.** *Let $X = [-1, 1]$ and $T = 3$. No finite-dimensional model from Definition 4.1 can uniformly approximate*

$$G_3(x_1, x_2, x_3) = x_1 x_2 \tag{32}$$

with error strictly smaller than 1 *on* $[-1, 1]^3$.

Proof. At time 3,

$$z_3 = A^2\beta(x_1) + A\beta(x_2) + \beta(x_3). \tag{33}$$

Thus, for each fixed value of x_3, the output can be written as

$$y_3(x_1, x_2, x_3) = u_{x_3}(x_1) + v_{x_3}(x_2) + w(x_3) \tag{34}$$

for suitable scalar functions u_{x_3}, v_{x_3}, and w. Consequently the mixed rectangle difference in the first two variables vanishes:

$$\Delta y_3(x_3) := y_3(1, 1, x_3) - y_3(1, -1, x_3) - y_3(-1, 1, x_3) + y_3(-1, -1, x_3) = 0. \tag{35}$$

For the target (32), however,

$$\Delta G_3 = 1 - (-1) - (-1) + 1 = 4. \tag{36}$$

If $\sup_{x_{1:3}\in[-1,1]^3} |y_3(x_{1:3}) - G_3(x_{1:3})| \le \delta$, then the four terms in the rectangle difference give

$$4 = |\Delta G_3 - \Delta y_3(x_3)| \le 4\delta. \tag{37}$$

Hence $\delta \ge 1$. Uniform error below 1 is impossible. □

The separation identifies the algebraic operation supplied by selectivity. A fixed additive transition accumulates transformed tokens, and its current-token readout reweights their additive representation. A selective coordinate can evolve as

$$h_1 = \phi_{j_1}(x_1), \qquad h_2 = \phi_{j_2}(x_2)h_1, \tag{38}$$

thereby creating the mixed feature $\phi_{j_1}(x_1)\phi_{j_2}(x_2)$ before any readout is applied. The tensor construction in Theorem 3.1 is a systematic finite-horizon version of this operation.

Clock-dependent input and readout maps The separation extends to comparison models whose input and readout maps also receive the clock. Writing $\beta_t(x) = \beta(x, e_t)$, and similarly for γ_t, κ_t, gives

$$z_3 = A^2\beta_1(x_1) + A\beta_2(x_2) + \beta_3(x_3), \qquad y_3 = \gamma_3(x_3)^\top z_3 + \kappa_3(x_3).$$

For each fixed x_3, the output is additively separable in x_1, x_2. Its mixed rectangle difference equals zero, while that of x_1x_2 equals 4. The same four-point argument gives uniform approximation error at least 1 for every state dimension and fixed transition matrix. Thus the separation holds with clock access available to both classes.

5 Autonomous clocks and neural selector realization

The clock code in Definition 2.1 is useful for the one-layer construction because it lets the selectors distinguish the first, second, and later scan steps. A preliminary scalar state-space layer can generate a finite-horizon clock internally, after which the selective tensor-partition construction can be applied with clock selectors that are functions of this scalar state.

Definition 5.1 (Depth-two selective SSM stack)**.** A depth-two selective SSM stack is a triangular composition of two diagonal selective recurrent layers. The lower layer has state $c_t \in \mathbb{R}^p$ and computes

$$c_0 = 0, \qquad c_t = \lambda^{(1)}(x_t) \odot c_{t-1} + b^{(1)}(x_t), \tag{39}$$

with continuous selectors $\lambda^{(1)} : X \to (0,1)^p$ and $b^{(1)} : X \to \mathbb{R}^p$. The upper layer has state $h_t \in \mathbb{R}^N$ and continuous selectors

$$\Lambda : X \times \mathbb{R}^p \to (0,1)^N, \qquad B : X \times \mathbb{R}^p \to \mathbb{R}^N, \qquad D : X \times \mathbb{R}^p \to \mathbb{R}^{m\times N}, \tag{40}$$

which may depend on the current token and lower-layer state:

$$h_0 = 0, \qquad h_t = \Lambda(x_t, c_t) \odot h_{t-1} + B(x_t, c_t), \qquad \widehat{y}_t = D(x_t, c_t)h_t. \tag{41}$$

Here Λ has coordinatewise values in $(0,1)$, and for each fixed (x_t, c_t) the output map is linear in h_t. Thus each layer is diagonal and affine in its own previous state; the only added convention is that upper-layer selectors may use the lower-layer state at the same sequence index.

Removing the external clock Fix $0 < \rho < 1$ and define

$$c_0 = 0, \qquad c_t = \rho c_{t-1} + 1. \tag{42}$$

Then

$$c_t = \frac{1-\rho^t}{1-\rho}, \tag{43}$$

so $c_1, \dots, c_T$ are distinct. Because this is a finite set, there exist continuous bump functions

$$\alpha_r : \mathbb{R} \to [0,1], \qquad \alpha_r(c_s) = \mathbf{1}_{\{r=s\}}, \qquad r, s \in \{1, \dots, T\}, \tag{44}$$

with pairwise disjoint supports if desired. Replacing the one-hot event "$e_t = e_r$" in (22)-(27) by the scalar condition encoded by $\alpha_r(c_t)$ gives selectors that take the same values along the realized trajectory $c_1, \dots, c_T$.

More explicitly, for $q = (\ell, j_1, \ldots, j_\ell)$, one may set

$$B_q(x, c) = \alpha_1(c)\phi_{j_1}(x), \tag{45}$$

while the gate can be chosen as

$$\Lambda_q(x, c) = \frac{1}{2} + \sum_{r=2}^{\ell} \alpha_r(c) \left(\phi_{j_r}(x) - \frac{1}{2} \right). \tag{46}$$

If the supports of the α_r are disjoint, then (46) is always a convex interpolation between $1/2$ and one of the values $\phi_{j_r}(x)$, and hence remains strictly inside $(0, 1)$. At the realized time $t = r$, it equals the gate used in (23). The readout can be defined, for $q = (\ell, j_1, \ldots, j_\ell)$, by

$$D_{:,q}(x, c) = \alpha_\ell(c) F_\ell(\xi_{j_1}, \ldots, \xi_{j_\ell}). \tag{47}$$

At $c = c_t$, this selects exactly the coordinates of length t and assigns the zero vector to all other columns.

Corollary 5.2 (Autonomous-clock universality)**.** *For every compact $X \subset \mathbb{R}^d$, finite horizon T, continuous causal map $F : X^T \to (\mathbb{R}^m)^T$, and $\varepsilon > 0$, there is a depth-two selective SSM stack in the sense of Definition 5.1, with no externally supplied clock code, that uniformly approximates F to error ε on X^T.*

Proof. The first layer implements (42) by taking $p = 1$, $\lambda^{(1)}(x) = \rho$, and $b^{(1)}(x) = 1$. The second layer uses the current input together with the scalar clock value c_t and the selectors in (45)-(47). Along every length-T trajectory, these selectors agree with the clocked selectors from the proof of Theorem 3.1. Hence the same tensor-product states are produced and the same uniform approximation bound holds. □

Selector networks and exponential SSM gates Proposition A.1 gives a finite neural realization using selector networks from a uniformly dense family, with network size allowed to grow. On the compact input-clock domain, the functions

$$q_i(u) = \log \frac{1 - \lambda_i(u)}{\lambda_i(u)}, \qquad B_i^\star(u) = \frac{b_i(u)}{1 - \lambda_i(u)}$$

are continuous and bounded. Finite networks g_i, $\widetilde{B}_i$, and $\widetilde{C}$ approximate q_i, $B_i^\star$, and C, respectively. With $\mathrm{softplus}(z) = \log(1 + e^z)$, setting

$$\widetilde{\Delta}_i = \mathrm{softplus}(g_i), \qquad \widetilde{\lambda}_i = e^{-\widetilde{\Delta}_i}, \qquad \widetilde{b}_i = (1 - \widetilde{\lambda}_i)\widetilde{B}_i \tag{48}$$

preserves strictly contractive exponential gates and the scalar zero-order-hold input form.

The finite-horizon error bound is explicit. Use the max norm for states and the induced infinity norm for readout matrices. If the three induced selector errors are at most $\eta \le 1$, let $H = T(\sup_u \|b(u)\|_\infty + 1)$ and $L_C = \sup_u \|C(u)\|_\infty$. Then

$$\begin{aligned} \left\| h_t - \widetilde{h}_t \right\|_\infty &\le t(H + 1)\eta, \\ \|\widehat{y}_t - \widetilde{y}_t\|_2 &\le \sqrt{m}\,[L_C T(H + 1) + H]\eta \end{aligned} \tag{49}$$

for every input sequence and $1 \le t \le T$. Appendix A.3 proves the bound and the uniform convergence of the induced selectors. Constructing the ideal scan to accuracy $\varepsilon/2$ and choosing the selector accuracy so that the additional output error is below $\varepsilon/2$ yields the desired accuracy ε. Thus finite neural selectors achieve universal causal approximation with exponential transitions and a linear-in-state readout.

6 Relation to existing universal approximation theorems

We compare Theorem 3.1 with classical universal approximation. Related theory covers approximation rates (Barron, 1993; Yarotsky, 2017), operator approximation (Chen and Chen, 1995; Grigoryeva and Ortega, 2018), and Turing universality (Siegelmann and Sontag, 1995). We first note that Theorem 3.1 is different from applying a feedforward theorem after flattening the sequence. In a flattened

MLP proof, all cross-token interactions are created by a nonlinear feedforward network acting on the entire vector $(x_1, \ldots, x_T)$. Here the readout is linear in the recurrent state. The feature family capturing nonlinear interactions is generated before the readout, inside the scan, through the identities $h_t^{(t,j_1,\ldots,j_t)} = \prod_{s=1}^{t} \phi_{j_s}(x_s)$. Thus the proof identifies a sequence-model-specific mechanism rather than merely invoking a general-purpose universal approximator.

Recent work has substantially expanded sequence-specific universality theory. Transformers are universal approximators of sequence-to-sequence functions, and sparse, efficient, shallow, and broader transformer-type variants satisfy related universal approximation guarantees (Yun et al., 2020a,b; Alberti et al., 2023; Kajitsuka and Sato, 2024; Cheng et al., 2025). Other sequence architectures have been analyzed through finite-memory temporal convolutional nets, stacked SSMs with layer-wise nonlinearities, and in-context approximation by prompted Transformers or fully recurrent models (Hanson and Raginsky, 2019; Wang and Xue, 2023; Petrov et al., 2024a,b). Recent Mamba-specific work studies how sharing input and output selection maps across channels in S6 can limit universality, and how block-biased variants recover approximation power by using separate selectors on groups of channels and adding channel-specific, input-independent biases (Yu and Erichson, 2025). These results are complementary to ours: they rely on attention-style token mixing, convolutional windows, nonlinear inter-layer maps, prompting, or different S6 parameter-sharing assumptions, whereas Theorem 3.1 uses a selective diagonal scan with a linear-in-state readout.

The theorem also substantially differs from RNN universality. The recurrence has no nonlinear activation applied to an affine function of h_{t-1}. Its state update is affine in the previous state, with a diagonal linear part. Universality is obtained despite this restriction because the diagonal coefficients are token dependent. In other words, the expressive resource is not arbitrary hidden-state dynamics; it is selective multiplication of prefix features by current-token selectors.

Note also that the proof of the theorem is not an attention or Transformer simulation. No layer forms pairwise token-token scores, and no argument routes information through all-to-all mixing. The construction is strictly causal and recurrent: each tensor-product feature is formed by propagating a single scalar coordinate forward and multiplying it by one local partition value at each step. This is the computational pattern that gives selective SSMs their linear-time scan structure.

Finally, Proposition 4.2 separates the result from expressivity claims about fixed-transition additive SSMs such as classical linear state-space recurrences and structured variants with input-independent dynamics (Gu et al., 2022a). A fixed transition can accumulate transformed tokens, but without input-dependent multiplication it cannot generally form mixed prefix interactions. The universal approximation theorem therefore pinpoints the additional expressive power supplied by the selective component of Mamba-style models.

7 Conclusion

We proved a constructive universal approximation theorem for Mamba-style diagonal selective state-space scans on compact finite-horizon sequence domains. The proof shows that input-dependent diagonal gates can build tensor-product partition features over prefixes, allowing a readout that is linear in the recurrent state to approximate arbitrary continuous causal maps. This identifies selectivity itself as a sufficient source of expressive cross-token interaction, rather than an effect that must come from attention, nonlinear hidden-state dynamics, or a universal post-scan network.

The separation extends to fixed-transition additive SSMs with clock-dependent input and readout maps, and a preliminary selective SSM layer generates the clock internally. Appendix A gives the S6 correspondence and finite neural realization, while Appendix B develops exact residual depth-width tradeoffs and compression through structured coefficient tensors.

These results give a concrete principle for designing expressive recurrent memory: token-dependent multiplication composes local features into interactions over input histories. When sampled target tensors admit short factorizations with nonzero nonnegative input-mode factors, each product term becomes one recurrent coordinate, linking the structure of temporal interactions to smaller states with controlled approximation error. Scan cost is linear in sequence length at fixed model size, while the general tensor-partition construction uses width exponential in the horizon for a fixed partition size.

Selectivity turns recurrent memory into a mechanism for composing the past with the present. These local interactions suffice to approximate every continuous causal rule on a compact input domain.

Acknowledgments and Disclosure of Funding

This work was supported by JST and JSPS. There are no competing financial or commercial interests; this is theoretical work on the universal approximation power of selective state-space models.

References

Silas Alberti, Niclas Dern, Laura Thesing, and Gitta Kutyniok. Sumformer: Universal approximation for efficient transformers. In *Workshop on Topology, Algebra, and Geometry in Machine Learning*, PMLR 221:72–86, 2023.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*, 2015.

Andrew R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, 1993.

Tianping Chen and Hong Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks*, 6(4):911–917, 1995.

Jingpu Cheng, Ting Lin, Zuowei Shen, and Qianxiao Li. A unified framework for establishing the universal approximation of transformer-type architectures. In *Advances in Neural Information Processing Systems*, 38:129131–129163, 2025.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734, 2014.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Q. Davis, Afroz Mohiuddin, Łukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4):303–314, 1989.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning*, PMLR 235:10041–10071, 2024.

Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.

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. In *International Conference on Learning Representations*, 2023.

Ken-ichi Funahashi and Yuichi Nakamura. Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Networks*, 6(6):801–806, 1993.

Lyudmila Grigoryeva and Juan-Pablo Ortega. Universal discrete-time reservoir computers with stochastic inputs and linear readouts using non-homogeneous state-affine systems. *Journal of Machine Learning Research*, 19(24):1–40, 2018.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on Language Modeling*, 2024.

Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. HiPPO: Recurrent memory with optimal polynomial projections. In *Advances in Neural Information Processing Systems*, 33:1474–1487, 2020.

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. In *Advances in Neural Information Processing Systems*, 34:572–585, 2021.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022.

Albert Gu, Ankit Gupta, Karan Goel, and Christopher Ré. On the parameterization and initialization of diagonal state space models. In *Advances in Neural Information Processing Systems*, 35:35971–35983, 2022.

Ankit Gupta, Albert Gu, and Jonathan Berant. Diagonal state spaces are as effective as structured state spaces. In *Advances in Neural Information Processing Systems*, 35:22982–22994, 2022.

Joshua Hanson and Maxim Raginsky. Universal approximation of input-output maps by temporal convolutional nets. In *Advances in Neural Information Processing Systems*, 32:14071–14081, 2019.

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

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.

Tokio Kajitsuka and Issei Sato. Are transformers with one layer self-attention using low-rank weight matrices universal approximators? In *International Conference on Learning Representations*, 2024.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, PMLR 119:5156–5165, 2020.

Patrick Kidger and Terry Lyons. Universal approximation with deep narrow networks. In *Conference on Learning Theory*, PMLR 125:2306–2327, 2020.

Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, 1993.

Harsh Mehta, Ankit Gupta, Ashok Cutkosky, and Behnam Neyshabur. Long range language modeling via gated state spaces. In *International Conference on Learning Representations*, 2023.

Antonio Orvieto, Samuel L. Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. In *International Conference on Machine Learning*, PMLR 202:26670–26698, 2023.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Xingjian Du, Matteo Grella, Kranthi Kiran GV, Xuzheng He, Haowen Hou, Jiaju Lin, Przemysław Kazienko, Jan Kocoń, Jiaming Kong, Bartłomiej Koptyra, Hayden Lau, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Bolun Wang, Johan S. Wind, Stanisław Woźniak, Ruichong Zhang, Zhenyuan Zhang, Qihang Zhao, Peng Zhou, Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. RWKV: Reinventing RNNs for the transformer era. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14048–14077, 2023.

Aleksandar Petrov, Philip Torr, and Adel Bibi. Prompting a pretrained transformer can be a universal approximator. In *International Conference on Machine Learning*, PMLR 235:40523–40550, 2024.

Aleksandar Petrov, Tom A. Lamb, Alasdair Paren, Philip H. S. Torr, and Adel Bibi. Universal in-context approximation by prompting fully recurrent models. In *Advances in Neural Information Processing Systems*, 37:72061–72093, 2024.

Allan Pinkus. Approximation theory of the MLP model in neural networks. *Acta Numerica*, 8:143–195, 1999.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y. Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning*, PMLR 202:28043–28078, 2023.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

Anton M. Schäfer and Hans-Georg Zimmermann. Recurrent neural networks are universal approximators. *International Journal of Neural Systems*, 17(4):253–263, 2007.

Hava T. Siegelmann and Eduardo D. Sontag. On the computational power of neural nets. *Journal of Computer and System Sciences*, 50(1):132–150, 1995.

Jimmy T. H. Smith, Andrew Warrington, and Scott W. Linderman. Simplified state space layers for sequence modeling. In *International Conference on Learning Representations*, 2023.

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:2307.08621, 2023.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Computing Surveys*, 55(6), Article 109, 28 pages, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 30:5998–6008, 2017.

Shida Wang and Beichen Xue. State-space models with layer-wise nonlinearity are universal approximators with exponential decaying memory. In *Advances in Neural Information Processing Systems*, 36:74021–74038, 2023.

Dmitry Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural Networks*, 94:103–114, 2017.

Annan Yu and N. Benjamin Erichson. Block-biased Mamba for long-range sequence processing. In *Advances in Neural Information Processing Systems*, 38:120541–120577, 2025.

Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? In *International Conference on Learning Representations*, 2020.

Chulhee Yun, Yin-Wen Chang, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. $O(n)$ connections are expressive enough: Universal approximability of sparse transformers. In *Advances in Neural Information Processing Systems*, 33:13783–13794, 2020.

A Mamba/S6 correspondence and finite neural selectors

The selective scan in Definition 2.1 isolates the input-dependent affine recurrence used by S6. This appendix gives its dimensional correspondence with the real-valued Mamba-1 scan, describes the surrounding mixer, and makes the finite neural realization explicit. We first describe the projection structure, channel sharing, and timestep parameterization of stock Mamba/S6. We then establish finite neural realizations for the selector class of Definition 2.1 using uniformly dense network families with increasing approximation capacity.

A.1 Dimensions, selectors, and discretization

Suppress the batch dimension. Let D be the number of inner channels, n the state dimension per channel, and r_Δ the intermediate timestep-projection dimension. At scan step t, write

$$v_t \in \mathbb{R}^D, \qquad H_t \in \mathbb{R}^{D\times n}, \qquad A \in (-\infty, 0)^{D\times n}.$$

Here v_t is the input to the scan, H_t is its recurrent state, and A lists the diagonal continuous-time generators for the D channels. The real-valued Mamba-1 parameterization takes the form

$$B_t = W_B v_t \in \mathbb{R}^n, \qquad C_t = W_C v_t \in \mathbb{R}^n, \tag{50}$$

$$\Delta_t = \mathrm{softplus}(W_{\Delta,2} W_{\Delta,1} v_t + a_\Delta) \in (0,\infty)^D, \tag{51}$$

where $W_B, W_C \in \mathbb{R}^{n\times D}$, $W_{\Delta,1} \in \mathbb{R}^{r_\Delta \times D}$, $W_{\Delta,2} \in \mathbb{R}^{D\times r_\Delta}$, and $a_\Delta \in \mathbb{R}^D$. The softplus function is $\mathrm{softplus}(z) = \log(1+e^z)$ and acts componentwise. The vectors B_t and C_t are shared across channels; the linear map inside the timestep selector has rank at most r_Δ. These are the parameter-sharing and projection conventions described in Gu and Dao (2024, Algorithm 2 and Section 3.6).

For exact zero-order-hold discretization, define

$$\overline{A}_{t,dj} = e^{\Delta_{t,d} A_{dj}}, \qquad \overline{B}^{\mathrm{ZOH}}_{t,dj} = \frac{e^{\Delta_{t,d} A_{dj}} - 1}{A_{dj}} B_{t,j}. \tag{52}$$

The corresponding state update and core scan output are

$$H_{t,dj} = \overline{A}_{t,dj} H_{t-1,dj} + \overline{B}_{t,dj} v_{t,d}, \qquad s_{t,d} = \sum_{j=1}^{n} C_{t,j} H_{t,dj}, \qquad H_0 = 0. \tag{53}$$

The official Mamba-1 implementation uses the same exponential transition and the input discretization

$$\overline{B}^{\mathrm{impl}}_{t,dj} = \Delta_{t,d} B_{t,j}. \tag{54}$$

Equations (52) and (54) specify the two input-discretization conventions explicitly. Both give a diagonal affine selective scan through (53). The implementation correspondence follows the reference selective-scan routine and the Mamba-1 module in the authors' code (Gu and Dao, 2024).[1]

To match Definition 2.1, flatten the state using the index (d, j), so $h_t \in \mathbb{R}^{Dn}$ with $(h_t)_{(d,j)} = H_{t,dj}$. Set

$$\lambda_{(d,j)}(v_t) = e^{\Delta_{t,d} A_{dj}}, \qquad b_{(d,j)}(v_t) = \overline{B}_{t,dj} v_{t,d}, \tag{55}$$

$$R(v_t)_{d,(e,j)} = \mathbf{1}_{\{d=e\}} C_{t,j}, \qquad R(v_t) \in \mathbb{R}^{D \times Dn}. \tag{56}$$

Then $h_t = \lambda(v_t) \odot h_{t-1} + b(v_t)$ and $s_t = R(v_t) h_t$. The matrix R corresponds to the readout selector called C in Definition 2.1; the separate notation distinguishes it from the channel-shared vector C_t above. Every coordinate of λ lies in $(0, 1)$. This identifies the core S6 recurrence with a structured instance of the generalized selective scan. Clock conditioning in the theorem is represented by the input $u_t = (x_t, e_t)$, or by the internally generated clock in Corollary 5.2.

A.2 The surrounding Mamba-1 mixer

Let $a_t \in \mathbb{R}^{d_{\text{model}}}$ be the representation entering a mixer. Its input projection produces two D-dimensional branches, and its output projection returns to the model dimension:

$$\begin{pmatrix} p_t \\ g_t \end{pmatrix} = W_{\text{in}} a_t + b_{\text{in}}, \qquad v_t = \mathrm{SiLU}\big((\mathrm{CausalConv}\, p)_t\big), \tag{57}$$

$$o_t = W_{\text{out}}\big[\mathrm{SiLU}(g_t) \odot \big(s_t + d_{\text{skip}} \odot v_t\big)\big] + b_{\text{out}}. \tag{58}$$

Here $W_{\text{in}} \in \mathbb{R}^{2D \times d_{\text{model}}}$, $W_{\text{out}} \in \mathbb{R}^{d_{\text{model}} \times D}$, $d_{\text{skip}} \in \mathbb{R}^D$, and $\mathrm{SiLU}(z) = z/(1 + e^{-z})$. Biases may be set to zero according to the chosen implementation. The causal convolution acts separately on each channel. Normalization and residual connections organize these mixers into the full sequence architecture.

The scan input v_t therefore includes the local causal preprocessing of the mixer. Equations (55)–(56) describe the correspondence at this scan interface. The skip term and output gate in (58) give an affine readout of the scan state for fixed branch inputs, while the theorem analyzes the linear-in-state readout of Definition 2.1. The latter readout combines tensor-product features that are already formed by the recurrence.

A.3 Finite parameters and range-preserving neural realization

For each fixed target and accuracy, the construction in Theorem 3.1 uses finitely many centers $\xi_1, \ldots, \xi_M$, one Gaussian scale, and the finite vector tables $F_t(\xi_{j_1}, \ldots, \xi_{j_t})$. These quantities specify all of its selectors. The following realization also expresses the selectors using finite feedforward networks while preserving the exponential gate form.

Let $\mathcal{U} = X \times \mathcal{E}_T$ with X nonempty and compact. Assume the chosen family of finite selector networks is uniformly dense in $C(\mathcal{U}, \mathbb{R}^k)$ for every finite k, with network size allowed to grow. Standard universal-approximation results provide such families (Cybenko, 1989; Hornik, 1991; Leshno et al., 1993; Pinkus, 1999; Kidger and Lyons, 2020).

Proposition A.1 (Neural realization with strictly contractive gates)**.** *Let (λ, b, C) be continuous selectors of a width-N scan on $\mathcal{U}$, with $\lambda_i(u) \in (0, 1)$ for every i, u. For every $\tau > 0$, finite selector networks g_i, $\widetilde{B}_i$, and $\widetilde{C}$ can be chosen so that the scan with*

$$\widetilde{\Delta}_i = \mathrm{softplus}(g_i), \qquad \widetilde{\lambda}_i = e^{-\widetilde{\Delta}_i}, \qquad \widetilde{b}_i = (1 - \widetilde{\lambda}_i)\widetilde{B}_i \tag{59}$$

and readout $\widetilde{C}$ has uniform output error less than τ relative to the original scan over the horizon T. Its gates lie strictly in $(0, 1)$ wherever the networks are evaluated.

Proof. Compactness and continuity give a uniform positive distance between each gate and both endpoints of $(0, 1)$. Consequently

$$q_i(u) = \log \frac{1 - \lambda_i(u)}{\lambda_i(u)}, \qquad B_i^\star(u) = \frac{b_i(u)}{1 - \lambda_i(u)} \tag{60}$$

[1] Official implementation: `selective_scan_interface.py` and `mamba_simple.py`.

are continuous and bounded on $\mathcal{U}$. The identity $e^{-\operatorname{softplus}(q_i)} = (1+e^{q_i})^{-1} = \lambda_i$ shows that uniform network approximations to q_i, $B_i^\star$, and C give uniform convergence of all three induced selectors to (λ, b, C). Each approximating gate lies in $(0,1)$ by its parameterization. The input term in (59) is the scalar zero-order-hold form with generator -1 and a constant input channel.

For completeness, use the hidden-state max norm and the induced readout matrix norm, and choose the selector approximations so that all three uniform errors are at most $\eta \le 1$. Writing $B_0 = \sup_{u\in\mathcal{U}} \|b(u)\|_\infty$, both hidden trajectories have norm at most $H = T(B_0+1)$ up to time T. If $E_t = \left\| h_t - \widetilde{h}_t \right\|_\infty$, then

$$E_t \le E_{t-1} + (H+1)\eta \le t(H+1)\eta.$$

With $L_C = \sup_u \|C(u)\|_\infty$, the readout error satisfies

$$\left\| C(u_t)h_t - \widetilde{C}(u_t)\widetilde{h}_t \right\|_\infty \le \big[L_C T(H+1) + H\big]\eta.$$

Multiplying this bound by $\sqrt{m}$ gives the Euclidean output bound. Choosing η sufficiently small proves the assertion. □

Applying Proposition A.1 to a tensor-partition scan with approximation error below $\varepsilon/2$ and taking $\tau = \varepsilon/2$ gives total approximation error below ε. Together with Theorem 3.1, this supplies finite neural selectors for universal causal approximation, preserving the selective multiplicative mechanism and exponential gates.

B Depth-width tradeoffs and structured tensor compression

This appendix develops two realizations of the tensor-partition approximant: an exact distribution of recurrent coordinates across residual depth, and a smaller product representation for coefficient tensors with suitable low-rank structure. The first construction applies to every scan in Definition 2.1; the second exploits structure in the sampled target values. Throughout this appendix, X is nonempty and compact.

B.1 Construction size and exact residual realization

With $M \ge 2$ labeled partition centers, Theorem 3.1 uses

$$N = \sum_{t=1}^{T} M^t = \frac{M^{T+1} - M}{M-1}. \tag{61}$$

For fixed M, this is $\Theta(M^T)$ as the horizon grows. The value of M is chosen from the target's continuity modulus and the requested accuracy, and may itself depend on the horizon. Equation (61) is the explicit state-size upper bound supplied by the constructive proof. For fixed width, the coordinatewise recurrent updates take $O(TN)$ arithmetic operations and maintain N recurrent scalars. Selector and readout evaluation add their own costs. This gives linear scan scaling in sequence length at fixed model size.

We use a residual architecture that preserves the current input-clock pair $u_t = (x_t, e_t)$ through depth and adds each block's readout to a pointwise output accumulator. This specifies the information available to each block in the following construction.

Proposition B.1 (Exact distribution of recurrent coordinates)**.** *Let a width-N diagonal selective scan with $N \ge 1$ be given, and let $W \ge 1$ be an integer. There is a residual realization with $L = \lceil N/W \rceil$ selective blocks, each having recurrent width at most W, that produces exactly the same output sequence. The sum of the recurrent widths is N.*

Proof. Partition $\{1, \dots, N\}$ into nonempty disjoint sets $I_1, \dots, I_L$ of size at most W. In block ℓ, set

$$h_t^{[\ell]} = \lambda_{I_\ell}(u_t) \odot h_{t-1}^{[\ell]} + b_{I_\ell}(u_t), \qquad h_0^{[\ell]} = 0. \tag{62}$$

The diagonal recurrence gives $h_t^{[\ell]} = (h_t)_{I_\ell}$ by induction on t. At each position initialize $a_t^{[0]} = 0 \in \mathbb{R}^m$ and propagate the residual stream $(u_t, a_t^{[\ell]})$ through depth by

$$a_t^{[\ell]} = a_t^{[\ell-1]} + C_{:,I_\ell}(u_t) h_t^{[\ell]}. \tag{63}$$

It follows that $a_t^{[L]} = \sum_{\ell=1}^{L} C_{:,I_\ell}(u_t)(h_t)_{I_\ell} = C(u_t)h_t$. The partition gives the claimed state sizes. □

Here W counts recurrent state coordinates per block. The preserved input-clock pair and the m-dimensional accumulator occupy the pointwise residual stream. With one-hot clocks its dimension is $d + T + m$; the scalar clock of Section 5 gives a corresponding stream of dimension $d + 1 + m$, with one additional scalar recurrent clock state. The accumulator runs across depth at each position. Thus the construction trades maximum recurrent width per block for depth while preserving the total feature-state count and the approximation error.

B.2 Low-rank coefficient tensors and positive selective gates

Recall the strictly positive partition functions ϕ_j from (14), with $\sum_j \phi_j(x) = 1$. For each prefix length t, denote the sampled vector coefficient tensor by

$$V^{(t)}_{j_1,\dots,j_t} = F_t(\xi_{j_1}, \dots, \xi_{j_t}) \in \mathbb{R}^m. \tag{64}$$

We consider a canonical polyadic (CP) factorization with nonnegative factors along the input modes and arbitrary real output vectors. This factor structure provides positive local features that can be realized by the gates.

Proposition B.2 (Selective realization of factored coefficients)**.** *For each $1 \le t \le T$, let $R_t \ge 1$ be an integer and suppose that*

$$\widetilde{V}^{(t)}_{j_1,\dots,j_t} = \sum_{r=1}^{R_t} a_{t,r} \prod_{s=1}^{t} v_{t,r,s,j_s}, \qquad a_{t,r} \in \mathbb{R}^m, \tag{65}$$

where every vector $v_{t,r,s} \in [0,\infty)^M$ is nonzero. Assume

$$\max_{j_1,\dots,j_t} \left\| V^{(t)}_{j_1,\dots,j_t} - \widetilde{V}^{(t)}_{j_1,\dots,j_t} \right\|_2 \le \delta_t. \tag{66}$$

Then a clocked diagonal selective scan of width $N_{\mathrm{fac}} = \sum_{t=1}^{T} R_t$, with gates in $(0,1)$ and a clock-dependent linear readout, realizes

$$\widetilde{Q}_t(x_{1:t}) = \sum_{j_1,\dots,j_t=1}^{M} \widetilde{V}^{(t)}_{j_1,\dots,j_t} \prod_{s=1}^{t} \phi_{j_s}(x_s). \tag{67}$$

Relative to the approximant Q_t in (19)*, its additional uniform error is at most δ_t at time t.*

Proof. Choose $K_{t,r,s} > \max_j v_{t,r,s,j}$ and define

$$g_{t,r,s}(x) = \frac{1}{K_{t,r,s}} \sum_{j=1}^{M} v_{t,r,s,j}\phi_j(x), \qquad \widehat{a}_{t,r} = a_{t,r} \prod_{s=1}^{t} K_{t,r,s}. \tag{68}$$

Strict positivity of the ϕ_j, the nonzero nonnegative factors, and their normalization give $0 < g_{t,r,s}(x) < 1$ for every $x \in X$. These functions are continuous.

Assign one state coordinate to each pair (t,r). At scan step k, choose its additive selector and gate as

$$b_{(t,r)}(x, e_k) = \begin{cases} g_{t,r,1}(x), & k = 1, \\ 0, & k \ne 1, \end{cases} \tag{69}$$

$$\lambda_{(t,r)}(x, e_k) = \begin{cases} g_{t,r,k}(x), & 2 \le k \le t, \\ 1/2, & \text{otherwise.} \end{cases} \tag{70}$$

Starting from zero state, induction gives $h_k^{(t,r)} = \prod_{s=1}^{k} g_{t,r,s}(x_s)$ for $1 \le k \le t$. At time $k = t$, read this coordinate with coefficient $\widehat{a}_{t,r}$; at all other times its readout column is zero. Expanding the

products then yields

$$\begin{aligned}\widehat{y}_t &= \sum_{r=1}^{R_t} \widehat{a}_{t,r} \prod_{s=1}^{t} g_{t,r,s}(x_s) \\ &= \sum_{r=1}^{R_t} a_{t,r} \prod_{s=1}^{t} \left(\sum_{j=1}^{M} v_{t,r,s,j} \phi_j(x_s) \right) = \widetilde{Q}_t(x_{1:t}). \end{aligned} \tag{71}$$

All gates are strictly contractive, and the readout depends only on the clock. Finally, the tensor-partition weights are nonnegative and sum to one, so

$$\begin{aligned}\left\| Q_t(x_{1:t}) - \widetilde{Q}_t(x_{1:t}) \right\|_2 &\le \sum_{j_1,\dots,j_t} \left\| V^{(t)}_{j_1,\dots,j_t} - \widetilde{V}^{(t)}_{j_1,\dots,j_t} \right\|_2 \prod_{s=1}^{t} \phi_{j_s}(x_s) \\ &\le \delta_t. \end{aligned} \tag{72}$$

Taking the supremum over prefixes proves the stated error bound. □

The factor condition permits zeros in the mode vectors: each factor needs at least one positive entry. Output vectors $a_{t,r}$ may have arbitrary signs. An identically zero coefficient tensor is represented by $R_t = 1$, $a_{t,1} = 0$, and all-ones mode vectors. More generally, one-hot mode vectors give the full coefficient-table representation, so the structured class includes the original construction. A factorization with $\sum_t R_t < \sum_t M^t$ gives a smaller recurrent realization at the stated coefficient accuracy. The sum $\sum_t R_t$ counts the terms in the supplied factorizations; the proposition applies directly to any such factorization.

B.3 Combining approximation, factorization, and residual depth

Write

$$E_{\mathrm{part}} = \max_{1 \le t \le T} \sup_{x_{1:t} \in X^t} \left\| F_t(x_{1:t}) - Q_t(x_{1:t}) \right\|_2, \qquad \delta = \max_{1 \le t \le T} \delta_t.$$

Proposition B.2 gives total approximation error at most $E_{\mathrm{part}} + \delta$ with N_{fac} recurrent coordinates. Applying Proposition B.1 distributes these coordinates among $\lceil N_{\mathrm{fac}}/W \rceil$ residual blocks of recurrent width at most W, with exactly the same outputs. For every $\tau > 0$, Proposition A.1 then gives a finite neural implementation with total error less than

$$E_{\mathrm{part}} + \delta + \tau. \tag{73}$$

For example, any choices satisfying $E_{\mathrm{part}} < \varepsilon/3$, $\delta \le \varepsilon/3$, and $\tau = \varepsilon/3$ achieve error below ε. The tensor-partition error, coefficient-factorization error, and neural selector error therefore have separate, explicit contributions.

Sparse or adaptive partitions offer additional ways to use target structure in the feature construction. Hierarchical and tensor-train factorizations provide a complementary direction: their successive contractions take the form $z_s = G_s(x_s) z_{s-1}$ with matrix-valued G_s. Implementing these contractions directly introduces input-dependent cross-channel mixing. Such selective extensions allow the recurrent computation to follow the target's compositional structure, alongside the diagonal positive-product construction proved above.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims match the results and assumptions in the paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Theorem 3.1, Section 3.3, and Appendices A–B specify the approximation setting, model parameterization, and construction costs.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: All theoretical results include assumptions and complete proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [N/A]

   Justification: The paper contains no experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [N/A]

Justification: The paper contains no experiments.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [N/A]

   Justification: The paper contains no experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [N/A]

   Justification: The paper contains no experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [N/A]

   Justification: The paper contains no experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work is theoretical, cites prior work, and involves no human subjects or sensitive data.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This is a foundational theory paper; there is no concern on societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: No data or models with high misuse risk are released.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: No existing research assets are used.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No new research assets are released.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: No crowdsourcing or human-subject research is involved.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No crowdsourcing or human-subject research is involved.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not an important, original, or non-standard part of the core methodology.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.