Interdomain Attention: Beyond Token-Level Key-Value Memory

Anonymous Authors[1]

Abstract

Transformers and deep SSMs sit at opposite ends of a basic design choice: attention learns where to read through query-dependent matching at quadratic cost and a growing KV cache, while deep SSMs compress context into a fixed-size recurrent state at the cost of query-independent read-out. We propose *Interdomain Attention*, which integrates an SSM into an attention module through kernel methods: an attention kernel is approximated by a finite feature map, the resulting key features and values are projected onto a shared set of basis functions maintained by a single SSM recurrence, and each query selects a slice of the stored coefficients through its own feature map, recovering the query-dependent read-out of attention at fixed state. The scalable layer is a learned relaxation of this derivation, and we validate its components through ablations. In a 125 M–1.3 B autoregressive language-modeling study on FineWeb-Edu at matched recurrent-state budget, Interdomain Attention improves on an SSM token mixer at every scale, surpasses a same-recipe softmax baseline at 1.3 B on validation perplexity and on the eight-task commonsense suite, and inherits the length-flat behavior of its fixed-state core out to $3.5\times$ the training context. Ablations indicate that the query-dependent readout is the main source of the gain.

1. Introduction

Softmax attention and state space models (SSMs) sit at opposite ends of a basic design choice for sequence models. Attention (Vaswani et al., 2017) keeps a per-token key-value cache and lets each query route through it by content-based matching, which gives sharp recall but costs $\mathcal{O}(N_q N)$ work and an $\mathcal{O}(N)$ KV state, where N_q is the query length and N is the key-value length. Hardware-aware kernels (Dao et al., 2022; Dao, 2024), KV caching (Shazeer, 2019), and distributed sharding (Shoeybi et al., 2019) mitigate but do not remove this scaling. Deep SSMs, building on HiPPO (Gu et al., 2020) and realized by S4 (Gu et al., 2022b), S4D (Gu et al., 2022a), and Mamba (Gu & Dao, 2024; Dao & Gu, 2024), take the opposite stance: they compress the entire context into a fixed-size recurrent state that updates in $\mathcal{O}(1)$ time per step. The cost of this compression is that the state is read out without conditioning on the query.

Hybrid architectures interleave the fine-grained, content-based retrieval of attention with the efficient long-range compression of SSMs (Lieber et al., 2024; Glorioso et al., 2024; Ren et al., 2025; De et al., 2024; Fu et al., 2023; Brixi et al., 2026), and a growing line of linear and sub-quadratic attention reduces the cost of attention directly (Katharopoulos et al., 2020; Peng et al., 2021; Poli et al., 2023; Peng et al., 2023; Sun et al., 2023; Yang et al., 2024a;b). We take a different route: rather than stacking a recurrent layer next to attention, we ask which attention-like read/write operations can be implemented inside a fixed-state recurrent memory. Holding the recurrent core to S4D and the per-token recurrent state to a fixed budget, we study what closes the gap between an S4D token mixer and a query-conditioned memory.

We answer this with *Interdomain Attention*, a token mixer in which keys and values are mapped to a shared SSM basis by a single complex S4D recurrence, and each query selects a slice of the resulting basis coefficients at readout. The construction is motivated by representing an attention kernel through a finite feature map and projecting the key features onto HiPPO basis functions, which yields a fixed-size state independent of sequence length. The scalable implementation is not a literal realization of the kernel derivation: it uses a learned SiLU/ℓ_2 feature map, input normalization, and a denominator-free readout. We therefore use the derivation as design motivation and evaluate the resulting layer empirically through ablations.

Our contributions are:

- A construction of a query-conditioned fixed-state to-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

ken mixer from a kernel-regression view of attention and a HiPPO-style basis projection, with an explicit boundary between the ideal derivation and the scalable implementation (Sections 3.1 and 3.3).

- A mechanism decomposition at 125 M parameters that separates the contributions of dual key/value write and query-dependent readout, and identifies the readout as the dominant axis (Appendix B.1).
- A 125 M–1.3 B iso-state language-modeling study on FineWeb-Edu in which Interdomain Attention improves over an S4D token mixer at every scale, surpasses a same-recipe softmax baseline on validation perplexity and the eight-task commonsense suite at 1.3 B, and preserves the length-flat behavior of the fixed-state core to $3.5\times$ the training context (Figure 2, Appendix B.4, Table 2).

2. Background

In this section, we briefly review attention mechanisms and state space models.

Attention as Kernel Regression. Standard dot-product attention (Vaswani et al., 2017) maps input tokens $x_n \in \mathbb{R}^d$ to queries, keys, and values via $q_n = W_q x_n \in \mathbb{R}^d$, $k_n = W_k x_n \in \mathbb{R}^d$, $v_n = W_v x_n \in \mathbb{R}^d$, and computes the output for the i-th token as

$$o_i = \frac{\sum_{n=1}^{N} \mathcal{K}(q_i, k_n)\, v_n}{\sum_{n'=1}^{N} \mathcal{K}(q_i, k_{n'})}, \tag{1}$$

where $\mathcal{K}(q,k) = \exp(q^\top k/\sqrt{d})$ for softmax attention. This is one realization of a Nadaraya-Watson kernel regression estimator (Nadaraya, 1964; Watson, 1964), a connection noted in several works on kernel attention (Tsai et al., 2019; Katharopoulos et al., 2020; Choromanski et al., 2021). This view makes the choice of kernel $\mathcal{K}$ a design degree of freedom: replacing the softmax kernel with one that admits a finite or learned feature representation enables sub-quadratic computation and memory. Attention then performs *nonparametric regression* over the values v_n at test time, with $\mathcal{K}$ determining the weighting over the context, connecting to the broader test-time regression (Wang et al., 2025) or memorization (e.g. Titans; Behrouz et al. 2025).

State Space Models and HiPPO. A linear state space model maps an input signal $z(t) \in \mathbb{R}$ to a latent state $u(t) \in \mathbb{R}^M$ via:

$$\dot{u}(t) = A(t)\, u(t) + B(t)\, z(t). \tag{2}$$

HiPPO (Gu et al., 2020) gives initializations for $A(t) \in \mathbb{R}^{M\times M}$ and $B(t) \in \mathbb{R}^M$ under which the state $u(t)$ maintains optimal projections of the input history onto M time-varying orthogonal basis functions $\{\phi_m^{(t)}\}_{m=1}^M$. This forms the foundation for deep SSM architectures such as S4 (Gu et al., 2022b), S5 (Smith et al., 2023), S4D (Gu et al., 2022a), and Mamba (Gu & Dao, 2024; Dao & Gu, 2024).

3. Interdomain Attention

Figure 1 illustrates the architecture. Building on the recently observed connection between interdomain kernel computation in HiPPO-SVGP and SSMs (Chen et al., 2025), we show how the kernel regression view Equation (1) leads to a direct integration of SSM-style recurrent memory into attention.

3.1. Feature-map view of kernel attention

Assume the attention kernel admits a finite feature representation

$$\mathcal{K}(q,k) \approx \xi(q)^\top \xi(k), \qquad \xi(\cdot) \in \mathbb{R}^R. \tag{3}$$

Substituting this representation into Equation (1) gives

$$o_i \approx \hat{o}_i = \frac{\sum_{n=1}^{N} \xi(q_i)^\top \xi(k_n)\, v_n}{\sum_{n'=1}^{N} \xi(q_i)^\top \xi(k_{n'})}. \tag{4}$$

Random Fourier features (Rahimi & Recht, 2007) are one standard instantiation for stationary kernels: by Bochner's theorem,

$$\begin{aligned} \mathcal{K}(x,x') &= \mathbb{E}_{p(\omega)}\Big[\xi_\omega(x)^\top \xi_\omega(x')\Big], \\ \xi_\omega(x) &= \big[\cos(\omega^\top x),\ \sin(\omega^\top x)\big]^\top, \end{aligned} \tag{5}$$

where $p(\omega)$ is the spectral density of $\mathcal{K}$ (its normalized Fourier transform). Approximating this expectation with sampled frequencies recovers Equation (3) with a cosine–sine feature map; the derivation below only uses the feature inner product.

3.2. HiPPO basis functions for interdomain attention

The key insight is to treat the keys and values as functions of time, $k(t_n) := k_n$ and $v(t_n) := v_n$, and project the key features, as well as the values, onto the HiPPO basis:

$$\begin{aligned} u_m^{(t_N)} &= \int \xi\big(k(t)\big)\, \phi_m^{(t_N)}(t)\, dt, \\ \gamma_m^{(t_N)} &= \int v(t)\, \phi_m^{(t_N)}(t)\, dt, \\ \eta_m^{(t_N)} &= \int \phi_m^{(t_N)}(t)\, dt, \end{aligned} \tag{6}$$

where $u_m^{(t_N)} \in \mathbb{R}^R$ and $\gamma_m^{(t_N)} \in \mathbb{R}^d$. We use u for these basis-projection coefficients to match the SVGP convention for interdomain inducing variables (Lázaro-Gredilla & Figueiras-Vidal, 2009; Hensman et al., 2013; Chen et al.,

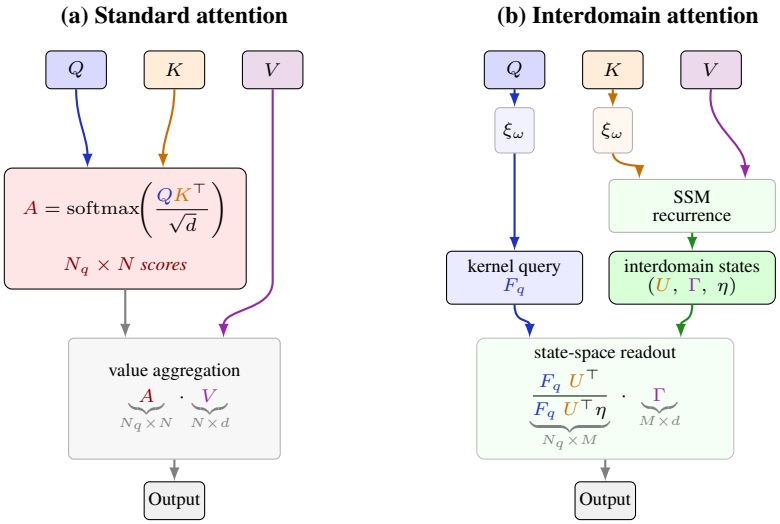


Figure 1. (a) Standard attention computes $N_q \times N$ scores from Q and K, then multiplies by V to produce the output. (b) Interdomain attention maps queries and keys into a shared feature space to produce the kernel query matrix F_q ($N_q \times R$). Keys and values are compressed via SSM recurrence into M interdomain states: U ($M \times R$, key feature projections), Γ ($M \times d$, value projections), and η ($M \times 1$, normalizing constants). State-space readout computes the output via the $N_q \times M$ cross-covariance $F_q U^\top$.

2025), with which the integrals above are in direct correspondence; in the practical S4D realization of Section 3.3, $u_m^{(t)}$ is replaced by a learned analogue rather than the literal integral. Both projections can be computed incrementally via the HiPPO ODE Equation (2) as new tokens arrive. Substituting the basis reconstruction $\xi(k_n) \approx \sum_m u_m\, \phi_m(t_n)$ into Equation (4) and exchanging the order of summation, we have:

$$\hat{o}_i \approx \frac{\sum_{m=1}^{M} \left(\xi(q_i)^\top u_m^{(t_N)}\right) \sum_{n=1}^{N} \phi_m^{(t_N)}(t_n)\, v_n}{\sum_{m=1}^{M} \left(\xi(q_i)^\top u_m^{(t_N)}\right) \sum_{n'=1}^{N} \phi_m^{(t_N)}(t_{n'})}, \tag{7}$$

Furthermore, recognizing $\sum_n \phi_m^{(t_N)}(t_n)\, v_n \approx \gamma_m^{(t_N)}$ and $\sum_{n'} \phi_m^{(t_N)}(t_{n'}) \approx \eta_m^{(t_N)}$ from Equation (6), the sums over tokens collapse (dropping the time superscript $^{(t_N)}$ for brevity):

$$\tilde{o}_i = \frac{\sum_{m=1}^{M} \left(\xi(q_i)^\top u_m\right) \gamma_m}{\sum_{m=1}^{M} \left(\xi(q_i)^\top u_m\right) \eta_m}, \tag{8}$$

In matrix form, let $F_q \in \mathbb{R}^{N_q \times R}$ be the query feature matrix, $U \in \mathbb{R}^{M \times R}$ the basis-projection (inducing-variable) matrix, $\Gamma \in \mathbb{R}^{M \times d}$ the value projection, and $\eta \in \mathbb{R}^M$. Then interdomain attention computes

$$\tilde{O} = \frac{F_q\, U^\top \Gamma}{F_q\, U^\top \eta}, \tag{9}$$

where division is element-wise with broadcasting. The entire context is summarized in U and Γ, which are updated recurrently. For causal processing, these become position-dependent: at step i, the SSM state encodes $U^{(i)}, \Gamma^{(i)}, \eta^{(i)}$ reflecting only tokens $1, \ldots, i$.

3.3. Our architecture

We assemble the components introduced above into a decoder-only language model for autoregressive language modeling.

Backbone. Interdomain Attention is embedded in a Llama-style pre-norm decoder (Touvron et al., 2023a) with RMSNorm (Zhang & Sennrich, 2019), SwiGLU feedforwards (Shazeer, 2020), RoPE (Su et al., 2024) on the query/key inputs, untied embeddings, and no dropout. The 1.3 B model uses d=2048, 24 layers, H=32 heads, head dimension d_h=64, context length $L_{\max}$=4096, S4D state dimension M=64, and feature dimension R=64. All heads keep their own keys and values (no grouped-query sharing).

Feature map. At language-model scale, we adopt the DeltaNet-style neural feature map (Yang et al., 2024b)

$$\xi(x) \;=\; \frac{\mathrm{SiLU}\left(\mathrm{SC}(x)\right)}{\left\|\,\mathrm{SiLU}\left(\mathrm{SC}(x)\right)\right\|_2}, \tag{10}$$

where $\mathrm{SC}(\cdot)$ is a causal depthwise 1D convolution of kernel size 4 applied to queries and keys before the head reshape, the same short-convolution primitive used in recent SSMs (Fu et al., 2023; Poli et al., 2023; Gu & Dao, 2024). With ξ defined via SiLU and ℓ_2-normalization, $\xi(q)^\top \xi(k)$ becomes a learned dot-product similarity in SiLU-projected space. This trains markedly more stably at scale, mirroring findings in DeltaNet (Yang et al., 2024b) and Gated Linear Attention (Yang et al., 2024a).

Recurrent basis projection. The HiPPO projections of Equation (6) are *approximated* by a complex-diagonal S4D recurrence (Gu et al., 2022a), per head:

$$x_t^{(h)} = \Lambda_h \odot x_{t-1}^{(h)} + B_h\, z_t^{(h)},$$
$$\Lambda_h = \exp(\Delta_h A_h) \in \mathbb{C}^M, \tag{11}$$

where A_h uses the S4D-Inv initialization (Gu et al., 2020; 2022a), $\log \mathrm{Re}(A_h) = \log \frac{1}{2}$ and $A_{h,\,\mathrm{imag}}(n) = \frac{M}{\pi}\left(\frac{M}{2n+1} - 1\right)$, and Δ_h is initialized log-uniformly in $[10^{-3}, 10^{-1}]$. This already departs from a literal implementation of Equation (6) in two respects: Λ_h is the diagonal approximation of the HiPPO state matrix introduced by S4D, and B_h together with the readout $C_h \in \mathbb{C}^{M\times M}$ (full complex per head, rather than the diagonal or identity variants) are *learned* rather than fixed to recover the HiPPO basis. We therefore treat the per-step outputs $U_h^{(t)}, \Gamma_h^{(t)}$ (complex-valued analogues of the matrices in Equation (9)) as a learned approximation of the basis-projection coefficients in Equation (8) rather than as their exact realization. The SSM input $z_t^{(h)}$ concatenates the key feature $\xi(k_t)$ and the value v_t; for the SiLU variant, both halves are first stabilized by an input RMSNorm with a learnable per-head bias (described next).

Input RMSNorm and denominator-free readout. The key feature and the value are independently passed through an RMSNorm with a learnable per-head additive bias before entering the SSM:

$$\tilde{k}_h^{(t)} = \mathrm{RMSNorm}\left(\xi(k_t^{(h)})\right) + b_h^k,$$
$$\tilde{v}_h^{(t)} = \mathrm{RMSNorm}(v_t^{(h)}) + b_h^v, \tag{12}$$

similar in spirit to B/C-side normalization used in Mamba-3 (Lahoti et al., 2026), with one difference: we normalize the two halves of the SSM *input* $z_t = [\xi(k_t), v_t]$ (the analogue of Mamba's B side), while the analogue of Mamba's C — the query-dependent read-out U_t in Equation (8) — is computed inside the SSM and is not separately normalized. The input rescaling acts on $\tilde{v}$ but not on the constant ones channel η, so retaining the Nadaraya–Watson denominator of Equation (8) would mix incompatible scales. We therefore drop the η-division and read out the *unnormalized* form

$$\tilde{O}_h \;=\; F_q^{(h)}\, U_h^\top\, \Gamma_h, \tag{13}$$

matching the denominator-free linear-attention convention shared by DeltaNet (Yang et al., 2024b), Mamba-2 (Dao & Gu, 2024), and GLA (Yang et al., 2024a). An optional SiLU output gate $o \leftarrow \sigma(W_g x) \odot o$ in the style of Mamba (Gu & Dao, 2024) and GLA (Yang et al., 2024a) is retained as a configuration flag but is disabled in the 1.3 B scaling runs.

Multi-head structure. Each of the $H{=}32$ heads owns its query/key/value projections, RMSNorm scales and biases of Equation (12), and S4D dynamics (Δ_h, A_h, C_h), yielding head-specific coefficients (U_h, Γ_h) in Equation (13). A grouped-KV variant that shares (U, Γ) across heads in the GQA (Ainslie et al., 2023) / MQA (Shazeer, 2019) style (equivalent to $n_{kv}{=}1$) is supported and reduces per-layer state by a factor of H; at the 1.3 B scale we use the fully per-head configuration. Causality is automatic: the per-token coefficients $U_h^{(i)}, \Gamma_h^{(i)}$ at position i depend only on tokens $1, \ldots, i$, so no explicit attention mask is used.

Training-time kernel. At training time, Equations (7) and (13) are evaluated through a fused Triton chunkwise kernel derived from the Flash Linear Attention algorithm introduced with GLA (Yang et al., 2024a): inside each chunk the intra-chunk contribution is expressed as a pair of matrix multiplications that map onto Tensor Cores, while cross-chunk state is propagated by a short sequential recurrence. This keeps the total work linear in sequence length at full-sequence quality; implementation details are in Appendix A.1.

From kernel regression to a learned relaxation. The recurrent basis projection of Equation (6) is itself approximated rather than realized exactly: S4D's diagonal Λ_h is the diagonalization of HiPPO-LegS dynamics, and B_h, C_h are learned end-to-end rather than fixed to recover the canonical HiPPO basis. On top of this, the SiLU variant uses a learned dot-product similarity (Equation (10)), input RMSNorm + bias on the SSM input (Equation (12)) which rescales the value branch, and the resulting denominator-free readout (Equation (13)). We therefore retain the kernel-regression view as design motivation and evaluate the practical layer empirically through the mechanism cube of Appendix B.1.

3.4. Memory and computational complexities

All complexities are per head, where N_q is the query length, N is the key-value length, M is the number of basis functions, R is the feature dimension, d is the head dimension, and K is the checkpoint interval.

Test-Time Generation. Per-step decode is $\mathcal{O}(M^2(R{+}d))$ for the full-rank C_h used here (or $\mathcal{O}(M(R{+}d))$ for diagonal C_h), independent of sequence length, against attention's $\mathcal{O}(Nd)$ work and growing KV cache. This makes per-token generation $\mathcal{O}(1)$ in N, a key advantage for long-context deployment.

4. Experiments

We evaluate Interdomain Attention in the regime where its fixed-state read/write structure is intended to matter most:

Table 1. Per-head computational and memory complexities. Interdomain attention replaces the $\mathcal{O}(Nd)$ KV cache with $\mathcal{O}(M(R{+}d))$ interdomain states, independent of sequence length N. K denotes the checkpoint interval (sequential scan) or chunk size (chunkwise scan).

	Total work	State memory	Backward memory
Standard attention	$\mathcal{O}(N_q N d)$	$\mathcal{O}(Nd)$	$\mathcal{O}(N_q N)$
Interdomain (FFT)	$\mathcal{O}(NM(R{+}d)\log N + N_q R d)$	$\mathcal{O}(M(R{+}d))$	$\mathcal{O}(NM(R{+}d) + N_q R)$
Interdomain (Scan)	$\mathcal{O}(NM(R{+}d) + N_q R d)$	$\mathcal{O}(M(R{+}d))$	$\mathcal{O}(\frac{N}{K}M(R{+}d) + N_q R)$

autoregressive language modeling. The study scales the architecture of Section 3.3 from 125 M to 1.3 B parameters on FineWeb-Edu at matched recurrent-state budget against an S4D token mixer, and reports a same-recipe softmax baseline as a reference point. The experiments are organized around three questions: whether the query-dependent readout explains the iso-state gain, whether the gain persists with scale, and how the fixed-state model behaves outside the training context.

4.1. Language modeling on FineWeb-Edu

Setup. We pretrain LLaMA-style decoder-only models (Touvron et al., 2023a) at four scales (125 M, 350 M, 760 M, 1.3 B parameters) on the FineWeb-Edu corpus (Penedo et al., 2024) with the Llama-2 tokenizer (Touvron et al., 2023b) (32K vocabulary) and a training context length of $L = 4096$. Each scale is trained at its Chinchilla-optimal token budget (Hoffmann et al., 2022) (approximately $20\times$ tokens per parameter): 2.5, 7, 15, and 26 billion tokens, respectively. We follow the training recipe of Gu & Dao (2024); full optimizer, schedule, and hardware details are in Appendix A.2. We report best-of-run validation perplexity from the cosine-decay schedule.

We compare four conditions, all sharing the same Llama-style backbone, dataset, tokenizer, and training recipe; only the token mixer in each block differs:

- *Softmax*: canonical multi-head softmax attention with rotary position embeddings;
- *Interdomain*: full mechanism of Section 3.3;
- *S4D-only*: an S4D control sharing Interdomain's complex S4D core (full per-head $C_h \in \mathbb{C}^{M\times M}$) but with both Interdomain ingredients removed: the dual semantic input $[\xi(k_t), v_t]$ is replaced by generic projections $[a_t, b_t]$, and the query-dependent read-out is replaced by a learned linear contraction of the SSM coefficient;
- *S4D-only + RoPE*: the S4D-only control with rotary position embeddings applied to the a-half (the K-side analogue), tested at 125 M only as a RoPE control.

Mechanism decomposition at 125 M. At the smallest scale a 3-axis ablation cube (dual write $\times$ Q-readout $\times$ RoPE) decomposes the iso-state gain. The query-dependent readout is the dominant axis: with the dual write retained, removing the Q-readout raises validation perplexity from 16.48 to 20.18 ($+22\%$ relative), close to the full S4D-only gap. The dual key/value write contributes a smaller $+2.8\%$ on its own ($16.48 \to 16.94$). RoPE is roughly orthogonal at this scale and mildly harmful inside the S4D family. The full cube is in Appendix B.1 and the per-condition data flow in Appendix B.2.

State-budget. Interdomain and the S4D-only variants share matched per-token recurrent state at every scale by construction (Appendix B.3); softmax is excluded from this fixed-state comparison since its KV cache grows linearly with sequence length.

Scaling. Figure 2 plots best-of-run FineWeb-Edu validation perplexity for the three main conditions across the four scales. The controlled iso-state comparison is Interdomain vs. S4D-only: Interdomain reaches 13–16% lower validation perplexity at every scale, indicating that the mechanism contribution persists as model size grows. Against the same-recipe softmax baseline, Interdomain is essentially tied at 125 M and pulls ahead from 350 M onwards, reaching 7.5% lower perplexity at 1.3 B (7.98 vs. 8.63); we treat the iso-state Interdomain vs. S4D-only gap as the controlled finding and the softmax comparison as a same-recipe reference point rather than a controlled one.

Downstream evaluation at 1.3 B. We evaluate the 1.3 B Softmax and Interdomain models via `lm-evaluation-harness` (Biderman et al., 2024) on the 8-task commonsense protocol of Yang et al. (2025) (LAMBADA (Paperno et al., 2016), PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2020), ARC-e/ARC-c (Clark et al., 2018), SIQA (Sap et al., 2019), BoolQ (Clark et al., 2019)), together with LAMBADA and WikiText-2 (Merity et al., 2017) language-modeling perplexities; the headline metrics and per-task breakdown are in the appendix (Sections B.4 and B.5). Relative to the same-recipe softmax baseline, Interdomain attention improves the commonsense 8-task average by $+3.03$ pp, the WikiText-2 BPB by -0.010, and the LAMBADA BPB by -0.131. The S4D control trails Softmax across the board: -2.07 pp on commonsense,

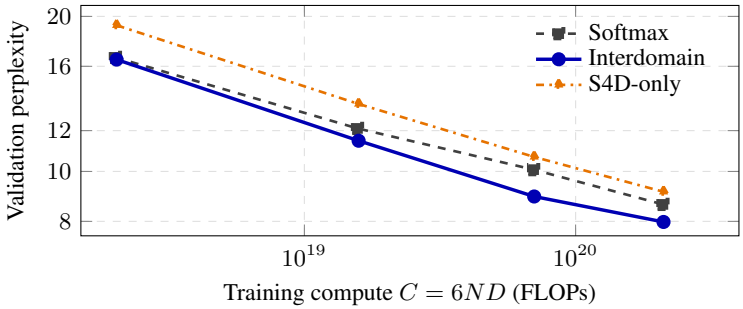


Figure 2. FineWeb-Edu validation perplexity vs. training compute (log–log scale), $C = 6ND$ FLOPs for total parameters N and total training tokens D (Chinchilla convention (Hoffmann et al., 2022)). Each point is best-of-run perplexity at the Chinchilla-optimal token budget for that scale (2.5, 7, 15, 26 B tokens for 125 M, 350 M, 760 M, 1.3 B parameters).

Table 2. Length-extrapolation perplexity at context length L, averaged over five long-context corpora (PG19, CodeParrot, GovReport, Qasper, QMSum). Training context is 4 K; 8 and 14 K are out-of-distribution. The per-corpus matrix is in Appendix B.6.

Context length	4 K (train)	8 K	14 K
Softmax	**12.79**	20.54	55.92
Interdomain	14.36	**14.14**	**14.26**
S4D-only	27.24	26.89	26.75

$\sim$2$\times$ LAMBADA perplexity (41.02 vs. 21.03), and $\sim$6% higher validation perplexity, consistent with the mechanism-cube finding (Appendix B.1) that removing both Interdomain ingredients regresses past softmax.

Length extrapolation. The fixed-state structure that Interdomain inherits from S4D summarizes the entire prefix in a state of size independent of context length, so the recurrence has no built-in dependence on training length. RoPE-based softmax attention, by contrast, degrades rapidly outside its training context. We evaluate the 1.3 B Softmax and Interdomain models at $L \in \{4\text{K}, 8\text{K}, 14\text{K}\}$ on five long-context corpora: PG19 (Rae et al., 2020), CodeParrot (Tunstall et al., 2022), GovReport (Huang et al., 2021), Qasper (Dasigi et al., 2021), and QMSum (Zhong et al., 2021). Within the 4 K training context softmax is slightly stronger than Interdomain (12.79 vs. 14.36), but softmax's average perplexity blows up beyond it (1.6$\times$ at 8 K, 4.4$\times$ at 14 K), while Interdomain stays within $\pm$0.25 of its 4 K value at every out-of-distribution length, a 3.5$\times$ extrapolation. We read length flatness as a property of the fixed-state recurrent core rather than of the Interdomain ingredients themselves: the S4D-only control is similarly length-flat (Table 2), and the mechanism contribution of Interdomain is best read off the validation-loss and downstream tables.

Recall and limitations. Exact-string associative recall is a known weak point of fixed-state token mixers, and Interdomain Attention is no exception (Yang et al., 2025; 2024b; Arora et al., 2024). On RULER single-needle-in-a-haystack (Hsieh et al., 2024), Phonebook exact-match retrieval (Jelassi et al., 2024), and the Based zero-shot recall suite (Arora et al., 2024) at 1.3 B (Appendix B.7), within the training context softmax dominates exact retrieval and Interdomain trails it, but the iso-state comparison still places Interdomain above the S4D control. Beyond the training context softmax collapses on RULER while Interdomain retains a small but non-zero score. The long-context LongBench-14 (Bai et al., 2024) downstream evaluation (Appendix B.8) is statistically tied between Softmax and Interdomain, with both well above S4D-only. We treat exact recall as a structural limitation of fixed-state compression rather than a property specific to Interdomain.

5. Conclusion and Future Work

We introduced Interdomain Attention, unifying kernel attention and state space models by projecting features of keys and values onto SSM basis functions via an SSM recurrence, yielding a fixed-size state independent of sequence length. In a 125 M–1.3 B FineWeb-Edu study at matched recurrent-state budget, Interdomain improves over an S4D token mixer at every scale, surpasses a same-recipe softmax baseline at 1.3 B on validation perplexity and the eight-task commonsense suite, and inherits the length-flat behavior of its fixed-state core out to 3.5$\times$ the training context. A 125 M mechanism decomposition attributes most of the iso-state gain to the query-dependent readout. In particular, combining the read/write structure introduced here with stronger recurrent cores such as Mamba-3 (Lahoti et al., 2026) and Gated DeltaNet (Yang et al., 2025) is one direction we leave for future work. Moreover, given the connection between kernel attention and Gaussian processes (Chen & Li, 2023), a probabilistic Interdomain Attention can be constructed by treating the layer as the posterior mean of an interdomain Gaussian process for improved robustness and uncertainty quantification (Chen, 2026).

References

Ainslie, J., Lee-Thorp, J., de Jong, M., Zemlyanskiy, Y., Lebrón, F., and Sanghai, S. GQA: training generalized multi-query transformer models from multi-head checkpoints. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 4895–4901. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.298.

Arora, S., Eyuboglu, S., Zhang, M., Timalsina, A., Alberti, S., Zou, J., Rudra, A., and Ré, C. Simple linear attention language models balance the recall-throughput tradeoff. In Salakhutdinov, R., Kolter, Z., Heller, K. A., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pp. 1763–1840. PMLR / OpenReview.net, 2024.

Bai, Y., Lv, X., Zhang, J., Lyu, H., Tang, J., Huang, Z., Du, Z., Liu, X., Zeng, A., Hou, L., Dong, Y., Tang, J., and Li, J. Longbench: A bilingual, multitask benchmark for long context understanding. In Ku, L., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 3119–3137. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.172.

Behrouz, A., Zhong, P., and Mirrokni, V. Titans: Learning to memorize at test time. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Biderman, S., Schoelkopf, H., Sutawika, L., Gao, L., Tow, J., Abbasi, B., Aji, A. F., Ammanamanchi, P. S., Black, S., Clive, J., DiPofi, A., Etxaniz, J., Fattori, B., Forde, J. Z., Foster, C., Hsu, J., Jaiswal, M., Lee, W. Y., Li, H., Lovering, C., Muennighoff, N., Pavlick, E., Phang, J., Skowron, A., Tan, S., Tang, X., Wang, K. A., Winata, G. I., Yvon, F., and Zou, A. Lessons from the trenches on reproducible evaluation of language models, 2024.

Bisk, Y., Zellers, R., Le Bras, R., Gao, J., and Choi, Y. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020. doi: 10.1609/aaai.v34i05.6239.

Brixi, G., Durrant, M. G., Ku, J., Poli, M., Brockman, G., Chang, D., Gonzalez, G. A., King, S. H., Li, D. B., Merchant, A. T., Naghipourfar, M., Nguyen, E., Ricci-Tam, C., Romero, D. W., Sun, G., Taghibakshi, A., Vorontsov, A., Yang, B., Deng, M., Gorton, L., Nguyen, N., Wang, N. K., Adams, E., Baccus, S. A., Dillmann, S., Ermon, S., Guo, D., Ilango, R., Janik, K., Lu, A. X., Mehta, R., Mofrad, M. R. K., Ng, M. Y., Pannu, J., Ré, C., Schmok, J. C., St. John, J., Sullivan, J., Zhu, K., Zynda, G., Balsam, D., Collison, P., Costa, A. B., Hernandez-Boussard, T., Ho, E., Liu, M.-Y., McGrath, T., Powell, K., Burke, D. P., Goodarzi, H., Hsu, P. D., and Hie, B. Genome modelling and design across all domains of life with Evo 2. *Nature*, 2026. doi: 10.1038/s41586-026-10176-5.

Chen, W. *Probabilistic Learning and Generation in Deep Sequence Models*. PhD thesis, Imperial College London, 2026.

Chen, W. and Li, Y. Calibrating transformers via sparse gaussian processes. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Chen, W., Kiyohara, N., Zhu, H. B. H., Curran-Sebastian, J., Bhatt, S., and Li, Y. Recurrent memory for online interdomain Gaussian processes. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Choromanski, K. M., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlós, T., Hawkins, P., Davis, J. Q., Mohiuddin, A., Kaiser, L., Belanger, D. B., Colwell, L. J., and Weller, A. Rethinking attention with performers. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Burstein, J., Doran, C., and Solorio, T. (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try ARC, the AI2 reasoning challenge, 2018.

Dao, T. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations*, 2024.

Dao, T. and Gu, A. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In Salakhutdinov, R., Kolter, Z., Heller, K.,

Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10041–10071. PMLR, 21–27 Jul 2024.

Dao, T., Fu, D. Y., Ermon, S., Rudra, A., and Ré, C. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 16344–16359. Curran Associates, Inc., 2022.

Dasigi, P., Lo, K., Beltagy, I., Cohan, A., Smith, N. A., and Gardner, M. A dataset of information-seeking questions and answers anchored in research papers. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tür, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y. (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pp. 4599–4610. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.NAACL-MAIN.365.

De, S., Smith, S. L., Fernando, A., Botev, A., Cristian-Muraru, G., Gu, A., Haroun, R., Berrada, L., Chen, Y., Srinivasan, S., Desjardins, G., Doucet, A., Budden, D., Teh, Y. W., Pascanu, R., Freitas, N. D., and Gulcehre, C. Griffin: Mixing gated linear recurrences with local attention for efficient language models, 2024.

Fu, D. Y., Dao, T., Saab, K. K., Thomas, A. W., Rudra, A., and Ré, C. Hungry hungry hippos: Towards language modeling with state space models. In *International Conference on Learning Representations*. OpenReview.net, 2023.

Glorioso, P., Anthony, Q., Tokpanov, Y., Whittington, J., Pilault, J., Ibrahim, A., and Millidge, B. Zamba: A compact 7B SSM hybrid model, 2024.

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

Gu, A., Dao, T., Ermon, S., Rudra, A., and Ré, C. Hippo: Recurrent memory with optimal polynomial projections. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1474–1487. Curran Associates, Inc., 2020.

Gu, A., Goel, K., Gupta, A., and Ré, C. On the parameterization and initialization of diagonal state space models. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 35971–35983. Curran Associates, Inc., 2022a.

Gu, A., Goel, K., and Ré, C. Efficiently modeling long sequences with structured state spaces. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022b.

Hensman, J., Fusi, N., and Lawrence, N. D. Gaussian processes for big data. In Nicholson, A. E. and Smyth, P. (eds.), *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, UAI 2013, Bellevue, WA, USA, August 11-15, 2013*. AUAI Press, 2013.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Vinyals, O., Rae, J. W., and Sifre, L. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, 2022.

Hsieh, C.-P., Sun, S., Kriman, S., Acharya, S., Rekesh, D., Jia, F., and Ginsburg, B. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*, 2024.

Huang, L., Cao, S., Parulian, N. N., Ji, H., and Wang, L. Efficient attentions for long document summarization. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tür, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y. (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pp. 1419–1436. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.NAACL-MAIN.112.

Jelassi, S., Brandfonbrener, D., Kakade, S. M., and Malach, E. Repeat after me: Transformers are better than state space models at copying. In Salakhutdinov, R., Kolter, Z., Heller, K. A., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pp. 21502–21521. PMLR / OpenReview.net, 2024.

Katharopoulos, A., Vyas, A., Pappas, N., and Fleuret, F. Transformers are rnns: Fast autoregressive transformers

with linear attention. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5156–5165. PMLR, 2020.

Kočiský, T., Schwarz, J., Blunsom, P., Dyer, C., Hermann, K. M., Melis, G., and Grefenstette, E. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018. doi: 10.1162/tacl_a_00023.

Lahoti, A., Li, K. Y., Chen, B., Wang, C., Bick, A., Kolter, J. Z., Dao, T., and Gu, A. Mamba-3: Improved sequence modeling using state space principles. In *The Fourteenth International Conference on Learning Representations*. OpenReview.net, 2026. Oral presentation.

Lázaro-Gredilla, M. and Figueiras-Vidal, A. R. Interdomain gaussian processes for sparse inference using inducing features. In Bengio, Y., Schuurmans, D., Lafferty, J. D., Williams, C. K. I., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada*, pp. 1087–1095. Curran Associates, Inc., 2009.

Lieber, O., Lenz, B., Bata, H., Cohen, G., Osin, J., Dalmedigos, I., Safahi, E., Meirom, S., Belinkov, Y., Shalev-Shwartz, S., Abend, O., Alon, R., Asida, T., Bergman, A., Glozman, R., Gokhman, M., Manevich, A., Ratner, N., Rozen, N., Shwartz, E., Zusman, M., and Shoham, Y. Jamba: A hybrid Transformer-Mamba language model, 2024.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Nadaraya, E. A. On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142, 1964. doi: 10.1137/1109020.

Paperno, D., Kruszewski, G., Lazaridou, A., Pham, Q. N., Bernardi, R., Pezzelle, S., Baroni, M., Boleda, G., and Fernández, R. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. doi: 10.18653/V1/P16-1144.

Penedo, G., Kydlícek, H., Allal, L. B., Lozhkov, A., Mitchell, M., Raffel, C. A., von Werra, L., and Wolf, T. The fineweb datasets: Decanting the web for the finest text data at scale. In Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J. M., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Peng, B., Alcaide, E., Anthony, Q., Albalak, A., Arcadinho, S., Biderman, S., Cao, H., Cheng, X., Chung, M., Derczynski, L., Du, X., Grella, M., Gv, K., He, X., Hou, H., Kazienko, P., Kocon, J., Kong, J., Koptyra, B., Lau, H., Lin, J., Mantri, K. S. I., Mom, F., Saito, A., Song, G., Tang, X., Wind, J., Woźniak, S., Zhang, Z., Zhou, Q., Zhu, J., and Zhu, R.-J. RWKV: Reinventing RNNs for the Transformer era. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 14048–14077, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.936.

Peng, H., Pappas, N., Yogatama, D., Schwartz, R., Smith, N. A., and Kong, L. Random feature attention. In *International Conference on Learning Representations*, 2021.

Poli, M., Massaroli, S., Nguyen, E., Fu, D. Y., Dao, T., Baccus, S., Bengio, Y., Ermon, S., and Ré, C. Hyena hierarchy: Towards larger convolutional language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 28043–28078. PMLR, 2023.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019.

Rae, J. W., Potapenko, A., Jayakumar, S. M., Hillier, C., and Lillicrap, T. P. Compressive transformers for long-range sequence modelling. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Rahimi, A. and Recht, B. Random features for large-scale kernel machines. In Platt, J., Koller, D., Singer, Y., and Roweis, S. (eds.), *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.

Ren, L., Liu, Y., Lu, Y., Shen, Y., Liang, C., and Chen, W. Samba: Simple hybrid state space models for efficient unlimited context language modeling. In *The Thirteenth*

International Conference on Learning Representations, ICLR 2025. OpenReview.net, 2025.

Sakaguchi, K., Le Bras, R., Bhagavatula, C., and Choi, Y. WinoGrande: An adversarial Winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8732–8740, 2020. doi: 10.1609/aaai.v34i05.6399.

Sap, M., Rashkin, H., Chen, D., Le Bras, R., and Choi, Y. Social IQa: Commonsense reasoning about social interactions. In Inui, K., Jiang, J., Ng, V., and Wan, X. (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4463–4473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1454.

Shazeer, N. Fast transformer decoding: One write-head is all you need, 2019.

Shazeer, N. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., and Catanzaro, B. Megatron-LM: Training multi-billion parameter language models using model parallelism, 2019.

Smith, J. T. H., Warrington, A., and Linderman, S. W. Simplified state space layers for sequence modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Su, J., Ahmed, M. H. M., Lu, Y., Pan, S., Bo, W., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2023.127063.

Sun, Y., Dong, L., Huang, S., Ma, S., Xia, Y., Xue, J., Wang, J., and Wei, F. Retentive network: A successor to Transformer for large language models, 2023.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. Llama 2: Open foundation and fine-tuned chat models, 2023b.

Tsai, Y.-H. H., Bai, S., Yamada, M., Morency, L.-P., and Salakhutdinov, R. Transformer dissection: An unified understanding for transformer's attention via the lens of kernel. In *EMNLP*, 2019.

Tunstall, L., von Werra, L., and Wolf, T. *Natural Language Processing with Transformers: Building Language Applications with Hugging Face*. O'Reilly Media, 2022. ISBN 9781098103248.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wang, K. A., Shi, J., and Fox, E. B. Test-time regression: a unifying framework for designing sequence models with associative memory. *arXiv preprint arXiv:2501.12352*, 2025.

Watson, G. S. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, 26(4):359–372, 1964.

Yang, S., Wang, B., Shen, Y., Panda, R., and Kim, Y. Gated linear attention transformers with hardware-efficient training. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 56501–56523. PMLR, 2024a.

Yang, S., Wang, B., Zhang, Y., Shen, Y., and Kim, Y. Parallelizing linear transformers with the delta rule over sequence length. In *Advances in Neural Information Processing Systems*, 2024b.

Yang, S., Kautz, J., and Hatamizadeh, A. Gated delta networks: Improving Mamba2 with delta rule. In *International Conference on Learning Representations*. OpenReview.net, 2025.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. HellaSwag: Can a machine really finish your sentence? In Korhonen, A., Traum, D., and Màrquez, L. (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800,

Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472.

Zhang, B. and Sennrich, R. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Zhong, M., Yin, D., Yu, T., Zaidi, A., Mutuma, M., Jha, R., Awadallah, A. H., Celikyilmaz, A., Liu, Y., Qiu, X., and Radev, D. R. Qmsum: A new benchmark for query-based multi-domain meeting summarization. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tür, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y. (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pp. 5905–5921. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.NAACL-MAIN.472.

A. Implementation and Training Details

A.1. SSM kernel backends

The FFT convolution computes the S4D recurrence (Gu et al., 2022b;a) via $\mathcal{O}(N \log N)$ transforms per input channel and state dimension. The sequential scan replaces the FFT with a fused recurrence that is $\mathcal{O}(N)$ per input channel and state dimension, with parallelism over channels and state dimensions on GPU. The chunkwise parallel scan, following the Flash Linear Attention scheme (Yang et al., 2024a), splits the sequence into chunks of size K. All chunks compute their terminal states in parallel, followed by a serial boundary propagation across N/K chunk boundaries, and a final parallel pass with corrected initial states. For the backward pass, both scan variants use segmented checkpointing with interval K: within each segment, previous states are recovered by inverting the diagonal SSM update (dividing by Λ_h). Loading the checkpoint at each segment boundary resets the numerical error that accumulates across this within-segment inverse recurrence. The parallel scan uses the parallel prefix algorithm (Smith et al., 2023), achieving $\mathcal{O}(\log N)$ parallel depth, which may become advantageous as hardware parallelism scales.

A.2. Language modeling training

A.2.1. Optimizer and schedule

Following Gu & Dao (2024): AdamW (Loshchilov & Hutter, 2019) with weight decay 0.1 and gradient clipping at 1.0; peak learning rate scaling with model size (3×10^{-3} at 125 M, 1.5×10^{-3} at 350 M, and 1×10^{-3} at both 760 M and 1.3 B), with a 375 M-token linear warmup followed by cosine decay to 10^{-5} over the remainder of training; global token batch size $524{,}288$ per optimization step (sequence-packed); SSM parameters (Equation (12), Δ_h, A_h, C_h) use a separate learning rate capped at 10^{-3} with weight decay disabled. Training uses bfloat16 mixed precision (`torch.autocast`) with random seed 42, on Isambard-AI GH200 nodes with 16–32-way DDP. The cosine decay to 10^{-5} leaves the best-of-run validation loss within 0.01 nats of the end-of-training loss for most cells.

A.2.2. Backbone details

SwiGLU hidden dimensions are set to $\frac{2}{3} \cdot 4d$ rounded up to a multiple of 128. Residual-path output projections (w_o of attention and w_2 of SwiGLU) are re-initialized with standard deviation $0.02/\sqrt{2L}$ following the GPT-style residual scaling rule (Radford et al., 2019).

A.2.3. Parameter counts

Table 3 reports the total trainable parameter count of every cell in the four-scale FineWeb-Edu sweep. The nominal scale labels (125 M, 350 M, 760 M, 1.3 B) are GPT-2-style shorthands; the Interdomain and S4D-only token mixers each add $\sim$0.5–1.0% parameters over the same-recipe softmax baseline (decreasing with scale, from $\sim 1.0\%$ at 125 M to $\sim 0.5\%$ at 1.3 B), due to the input RMSNorm scales and biases of Equation (12), the per-head (Δ_h, A_h, C_h) SSM dynamics, and the ShortConv. Within each iso-state row, Interdomain and S4D-only agree to within 0.01%. For the 125 M mechanism decomposition of Section B.1, all six Interdomain/S4D variants lie in $[135{,}379{,}344,\ 135{,}462{,}288]$, a 0.06% spread, with the softmax baseline at $134{,}105{,}856$.

Table 3. Total trainable parameters for every cell in the language-modeling scaling sweep. Counts include token + output embeddings (untied, 32,000-vocab Llama-2 tokenizer (Touvron et al., 2023b)), all attention/SSM weights, the input-RMSNorm scales and biases of Equation (12), the layer-level RMSNorms, and the SwiGLU feedforward.

Condition	125 M	350 M	760 M	1.3 B
Softmax	134,105,856	373,867,520	777,856,512	1,345,423,360
Interdomain	135,416,208	377,360,768	781,498,752	1,352,406,784
S4D-only	135,425,424	377,385,344	781,523,328	1,352,455,936

B. Language Modeling: Supplementary Material

This appendix expands the experiments of Section 4.1.

B.1. Mechanism decomposition cube at 125 M

We expand the 3-axis ablation cube of Section 4.1. The two Interdomain ingredients form binary axes: (i) the *dual input* $z_t = [\xi(k_t), v_t]$ that splits the SSM write into a key-feature half and a value half, and (ii) the *query-dependent readout* $\xi(q_t)\, U_t^\top\, \Gamma_t$ that lets the per-token query select a state slice. RoPE is a third, independent axis, tested at both endpoints of the Interdomain/S4D axis. Six of the eight cube corners, plus the softmax baseline, are reported in Table 4; Figure 3 shows the per-condition data flow.

Table 4. Mechanism decomposition at 125 M / 2.5 B tokens. "vs Softmax" is the relative change in validation perplexity (positive = worse than softmax). Val. loss is best-of-run; Val. PPL is $\exp(\text{Val. loss})$.

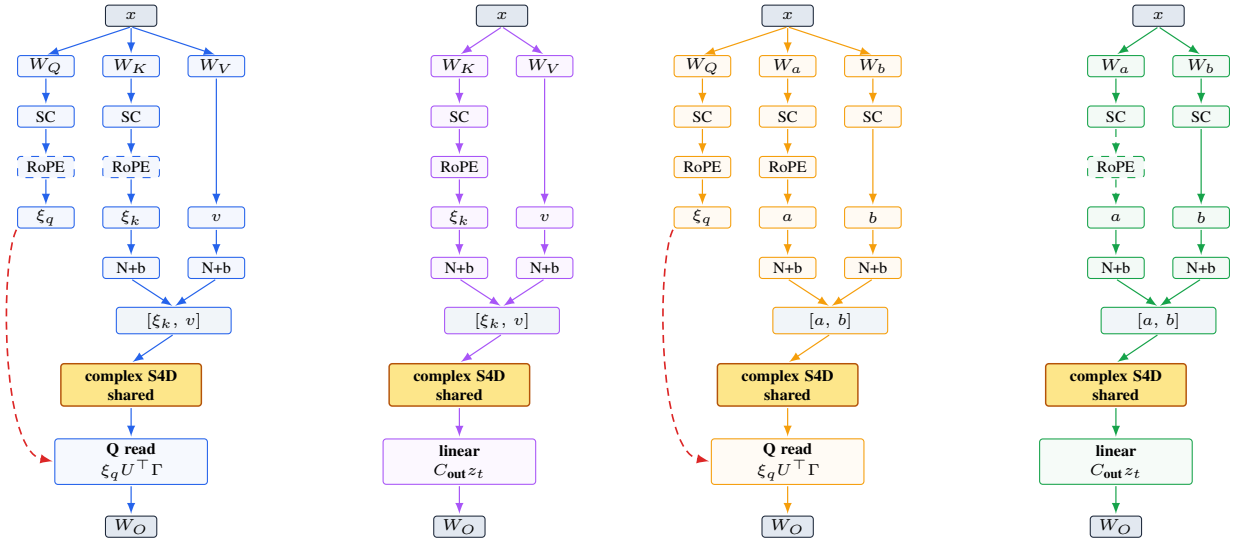
Condition	Description	Val. loss	Val. PPL	vs Softmax
Full Interdomain	Dual write + Q-readout	**2.8020**	**16.48**	**−1.02%**
Full, no RoPE	Full Interdomain without Q/K RoPE	2.8038	16.51	−0.84%
Softmax	Canonical Llama-style softmax baseline	2.8122	16.65	—
Single write + Q-readout	Single $[a, b]$ input + Q-readout	2.8298	16.94	+1.78%
S4D-only	Vanilla S4D, no RoPE	2.9561	19.22	+15.48%
S4D-only + RoPE	Vanilla S4D, RoPE on a-half	2.9844	19.77	+18.79%
Dual write + linear readout	Dual input $[\xi(k), v]$ without Q	3.0048	20.18	+21.23%

B.1.1. Reading the cube

The headline contrast is Interdomain versus the S4D-only control at matched recurrent-state budget: Full Interdomain reduces validation perplexity from 19.22 to 16.48, a 14.3% relative reduction at iso-state. Beyond the headline finding (Q-readout dominant; RoPE orthogonal) reported in Section 4.1, the cube also pins down two cells the body does not: keeping the Q-readout while feeding the SSM a generic $[a, b]$ input recovers most of softmax-level performance (+1.7% relative to softmax, within 0.3 PPL of it), confirming that the dual key/value write is a smaller secondary contributor; and removing Q/K RoPE from Full Interdomain leaves it within noise of the full model, suggesting that Interdomain's mechanism already encodes the positional selectivity that RoPE supplies to softmax attention. The pure-S4D vs. softmax gap of +15 to +19% at this scale is consistent with the SSM literature; Fu et al. (2023) report a comparable Pile-scale gap for pure S4, and FineWeb-Edu is an easier corpus on which our S4D-only variants additionally carry the ShortConv and pre-SSM-norm stabilizers shared with Interdomain.

B.2. Mechanism-cell architecture diagram

Figure 3 shows the per-condition data flow referenced in Section B.1. The four subfigures show the *Full Interdomain* mechanism, the *Dual write, linear read* variant, the *Single write, Q read* variant, and the *S4D-only* control.



(a) Full Interdomain *(b)* Dual write, linear read *(c)* Single write, Q read *(d)* S4D-only

Figure 3. Architecture across the four mechanism cells. **(a)** *Full Interdomain* uses both the dual write $z_t = [\xi(k_t), v_t]$ and the Q-mediated readout $\xi(q_t)U_t^\top \Gamma_t$. **(b)** *Dual write, linear read* keeps the dual write but removes Q from the readout, replacing it with a learned linear contraction. **(c)** *Single write, Q read* feeds the SSM a generic two-half input $[a_t, b_t]$ while retaining the Q-mediated readout. **(d)** *S4D-only* removes both Interdomain ingredients. Branches drawn in the subfigure colour are active; the dashed red curve marks the Q bypass; the amber complex-valued S4D core is identical across all four variants. The two "N+b" boxes denote the per-head input RMSNorm + learnable bias of Equation (12), applied independently to each SSM input half. (RoPE is the third, orthogonal axis; see Table 4.)

B.3. State budget at 1.3 B

Table 5 summarizes the per-token recurrent state of the token mixers at the 1.3 B scale, following the state-DoF accounting convention of Lahoti et al. (2026, Prop. 2): complex SSM state of dimension N counts as $2N$ real DoF. The Interdomain SSM ingests a two-half input $z_t = [\xi(k_t), v_t]$ of width $d_{\text{feat}} + d_h$ ($= 2d_h$ in our configuration, where $d_{\text{feat}} = d_h$), so the per-cell state is $2(d_{\text{feat}} + d_h)M$ real DoF; the S4D-only variants likewise ingest a two-half $[a, b]$ input of the same width and inherit the same per-cell state. With matched head counts H and matched M, Interdomain and the S4D-only variants are *iso-state by construction*. Softmax attention is excluded from the fixed-state column because its KV cache grows linearly with sequence length L.

Table 5. Recurrent-state size at 1.3 B (d=2048, 24 layers, H=32, d_h=64, M=64). #cells is the number of independent state cells per layer; per-cell DoF is the size of each cell's hidden state in real-valued degrees of freedom; total DoF is their product, reported per token per layer. RoPE does not change state size; the S4D-only + RoPE row is identical to S4D-only and is omitted.

Condition	#cells	per-cell DoF	total DoF
Softmax	—	KV cache (grows with L)	4,096 L
Interdomain	$H = 32$	$2(d_{\text{feat}}{+}d_h)\,M = 16{,}384$	**524,288**
S4D-only	$H = 32$	$2(d_h{+}d_h)\,M = 16{,}384$	**524,288**

B.4. Downstream evaluation at 1.3 B

Table 6. Validation cross-entropy and downstream metrics for the 1.3 B / 26 B-token runs. LAMBADA (Paperno et al., 2016) and WikiText-2 (Merity et al., 2017) are evaluated zero-shot via `lm-evaluation-harness` (Biderman et al., 2024). BPB (bits per byte) rescales the per-token cross-entropy by tokens/byte, yielding a tokenizer-invariant comparison.

Metric	Softmax	Interdomain	S4D-only
Validation CE ↓	2.155	**2.077**	2.212
Commonsense 8-avg ↑	51.51	**54.54**	49.44
LAMBADA acc ↑	39.43	**44.36**	25.64
LAMBADA PPL ↓	21.03	**14.76**	41.02
LAMBADA BPB ↓	1.1244	**0.9937**	1.3712
WikiText-2 PPL ↓	20.55	**19.77**	25.27
WikiText-2 BPB ↓	0.8156	**0.8051**	0.8713

B.5. Commonsense 8-task breakdown at 1.3 B

Table 7. Per-task accuracy of the eight-task commonsense suite of Yang et al. (2025), evaluated on the 1.3 B / 26 B-token runs via `lm-evaluation-harness` (Biderman et al., 2024). HellaSwag and ARC-c use length-normalized accuracy; all others use plain accuracy. The 8-task avg row corresponds to the headline column in Table 6.

Task	Softmax	Interdomain	S4D-only
LAMBADA (acc)	39.43	**44.36**	25.64
PIQA	69.42	**71.38**	71.00
HellaSwag (acc_norm)	48.53	**54.27**	49.34
WinoGrande	54.22	**57.85**	53.83
ARC-easy	65.87	**69.91**	60.86
ARC-challenge (acc_norm)	33.02	**37.46**	33.36
SIQA	40.28	**41.45**	41.10
BoolQ	**61.28**	59.63	60.37
8-task avg	51.51	**54.54**	49.44

B.6. Per-corpus length-extrapolation perplexity

Table 8. Per-corpus validation perplexity at context lengths $L \in \{4, 8, 14\}$ K for the 1.3 B Softmax, Interdomain, and S4D-only models on PG19 (Rae et al., 2020), CodeParrot (Tunstall et al., 2022), GovReport (Huang et al., 2021), NarrativeQA (Kočiský et al., 2018), Qasper (Dasigi et al., 2021), and QMSum (Zhong et al., 2021). Training context is 4 K; 8 K and 14 K are out-of-distribution. NarrativeQA is reported here only and is excluded from the five-corpus aggregate of Table 2.

Corpus	Condition	4 K	8 K	14 K
PG19	Softmax	**16.34**	26.78	70.95
	Interdomain	15.86	**15.73**	**15.94**
	S4D-only	19.96	19.75	19.66
CodeParrot	Softmax	**12.13**	22.23	74.85
	Interdomain	17.62	**17.31**	**17.48**
	S4D-only	53.10	52.44	52.16
GovReport	Softmax	**7.33**	12.20	31.76
	Interdomain	7.36	**7.27**	**7.29**
	S4D-only	9.19	9.07	9.01
NarrativeQA	Softmax	**21.28**	34.36	92.75
	Interdomain	22.85	**22.67**	**22.99**
	S4D-only	36.83	36.42	36.26
Qasper	Softmax	**14.43**	22.94	60.08
	Interdomain	15.75	**15.52**	**15.63**
	S4D-only	23.53	23.22	23.13
QMSum	Softmax	**13.74**	18.57	41.96
	Interdomain	15.20	**14.88**	**14.94**
	S4D-only	30.44	29.97	29.80

B.7. Recall-heavy tasks at 1.3 B

Table 9. RULER single-needle-in-a-haystack (Hsieh et al., 2024) (S-NIAH-1, passkey retrieval), Phonebook exact-match retrieval (Jelassi et al., 2024), and Based zero-shot recall (Arora et al., 2024) (*contains*, case-insensitive accuracy, %) at 1.3 B. Within the training context, recall-heavy tasks favour softmax attention; state-based architectures (Interdomain, S4D) are much weaker at exact-string retrieval. The Based sub-block reports the six-task suite (SWDE, FDA, SQuAD, NQ, TriviaQA, DROP) together with the 6-task average.

RULER S-NIAH-1 (accuracy %)				
Context length	1 K	2 K	4 K	8 K
Softmax	**99.50**	**100.00**	**89.00**	0.00
Interdomain	65.50	28.00	9.50	**3.50**
S4D-only	0.00	0.00	0.00	0.00
Phonebook exact-match retrieval (accuracy %)				
List size	16	32	64	128
Softmax	**54.50**	**52.00**	**45.50**	**24.00**
Interdomain	0.00	0.00	0.00	0.00
S4D-only	0.00	0.00	0.00	0.00
Based zero-shot recall (contains-CI accuracy %)				
Task	SWDE	FDA	SQuAD	NQ
Softmax	**60.58**	**50.45**	11.33	**16.44**
Interdomain	17.46	9.53	**33.78**	11.43
S4D-only	6.57	0.36	14.91	7.95
Task	TriviaQA	DROP	6-task avg	
Softmax	**56.93**	**23.47**	**36.53**	
Interdomain	54.27	21.31	24.63	
S4D-only	42.83	14.04	14.44	

B.8. LongBench downstream evaluation

Table 11. Steady-state decode latency summary at 1.3B. "Range" is the min–max over the prefix lengths that fit; "max-fit L" is the largest prefix length that ran without OOM or capture failure. Speedup is computed as the median softmax latency divided by the median graphed-interdomain latency. The graph-capture advantage shrinks monotonically with B; by B=64 the kernel-launch overhead is a small fraction of total decode time and graphed interdomain is slower than softmax in absolute terms (compute-bound regime).

	Softmax (SDPA)		Interdomain			
B	Range (ms)	Max-fit L	Eager (ms)	Graphed (ms)	Graphed max-fit L	Speedup
1	18.20–19.36	16,384	39.07–40.29	**8.06–8.08**	16,384	2.33×
8	18.71–19.25	16,384	40.91–41.97	**10.16–10.20**	16,384	1.86×
16	18.56–19.12	8,192	39.86–41.12	**12.63–12.65**	16,384	1.50×
32	19.00–19.31	4,096	40.90–41.54	17.32–17.35	8,192	1.10×
64	**18.94–19.68**	2,048	41.18–42.82	25.73–25.77	4,096	0.75×

We additionally evaluate the 1.3 B Softmax / Interdomain / S4D-only models on LongBench (Bai et al., 2024), using the 14-subtask configuration of Yang et al. (2025). Table 10 reports the 14-task average (LongBench scores normalized to $[0, 1]$).

Table 10. LongBench 14-task average score (range $[0, 1]$) at 1.3 B. Softmax and Interdomain are within 0.0011 of each other, while S4D-only trails by 0.038.

Condition	14-task avg ↑
Softmax	**0.1240**
Interdomain	0.1229
S4D-only	0.0857

C. Inference Scaling

C.1. Autoregressive Decode: Prefix-Length Scaling

We benchmark autoregressive decode latency for a 1.3B-parameter Llama-style model on a single NVIDIA GH200 (120 GB HBM3) using bfloat16 mixed precision (`torch.autocast`). The architecture matches the trained 1.3B of Section 4.1 (d=2048, 24 layers, H=32, d_h=64, M=64, 32,000-vocab Llama-2 tokenizer); decode is run on randomly initialized weights since latency depends only on shapes. We compare three code paths:

- **Softmax (SDPA).** Eager-mode Python decode loop with attention computed by PyTorch's `scaled_dot_product_attention`; on Hopper with recent PyTorch versions this dispatches to the cuDNN-FA backend.
- **Interdomain (eager).** Eager-mode Python decode loop with chunked prefill (C=2048).
- **Interdomain (graphed).** The same fixed-shape decode body captured into a single `torch.cuda.CUDAGraph` and replayed.

Within our test window ($B \leq 64$, $L \leq 16{,}384$), the softmax decode body is essentially prefix-invariant, so the comparison is not a prefix-length crossover. The interdomain advantage in this window comes from two structural factors:

1. **CUDA graph compatibility.** The fixed-shape SSM state lets the entire decode body be captured into a single static graph and replayed once per token, removing the per-step Python and kernel-launch overhead. Softmax decode requires a dynamically growing KV cache and is not directly graph-capturable; eliminating its launch overhead would require padded allocation or bucketed graph pools.
2. **Lower peak prefill memory via chunking.** Because the recurrent state has a size independent of prefix length, prefill can be processed in fixed-size chunks of C=2048 tokens with the running state retained and per-chunk activations released. This lets interdomain decode reach (B, L) cells where softmax exhausts GH200 HBM3 memory.

C.1.1. Protocol

Prefill L tokens → snapshot the KV/SSM state (and capture the CUDA graph for the graphed path) → 64 decode steps timed with CUDA events. Warmup: 5 iterations; timed: 20 iterations (15 at B=32, 10 at B=64); `compile=False` for both methods.

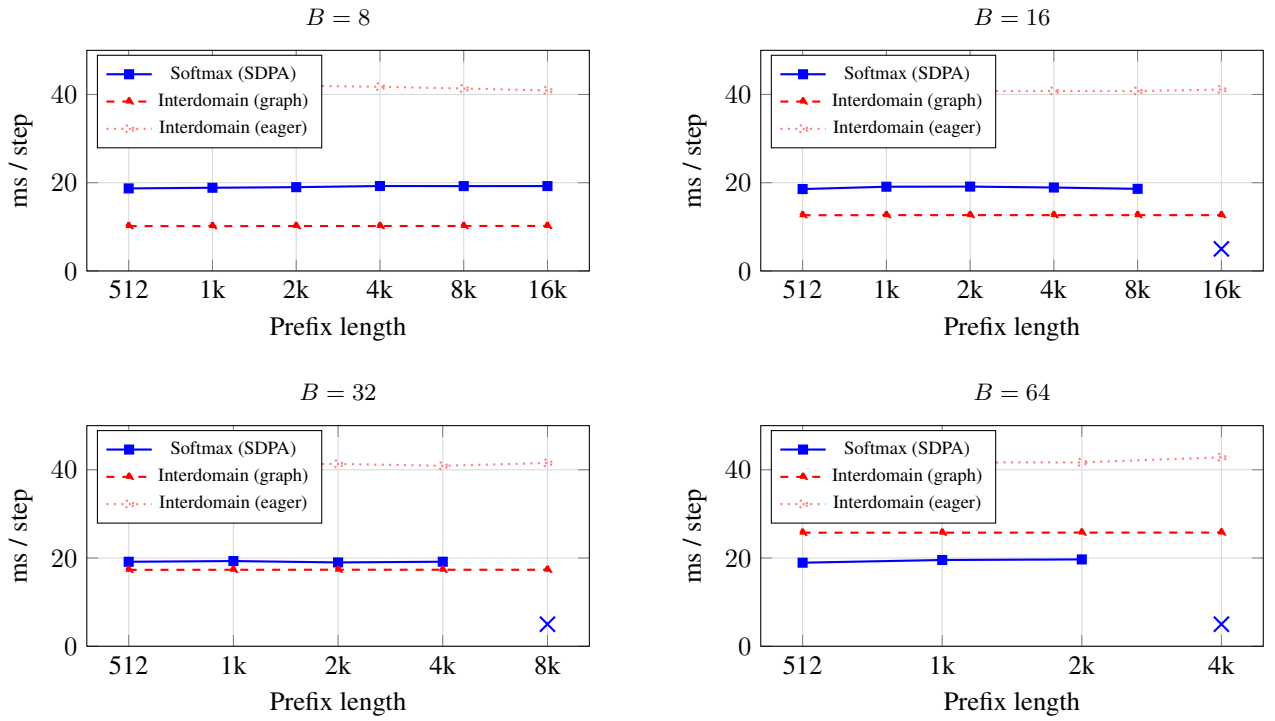


Figure 4. Steady-state decode latency vs. prefix length at varying batch sizes for the 1.3B model. Both softmax (solid blue) and graphed interdomain (dashed red) are essentially prefix-invariant across our test window; eager interdomain (dotted, light red) is also prefix-invariant but slower than the graphed path by an offset corresponding to the per-step kernel-launch overhead. × marks out-of-memory cells. Chunked prefill enables interdomain decode through L=16,384 at $B \leq 16$ and through L=8,192 at B=32, regions where softmax exhausts the 120 GB HBM3.

C.1.2. Latency-sensitive regime ($B = 1$)

At batch size 1, kernel-launch overhead dominates per-step latency. CUDA-graph capture brings interdomain decode from 39–40 ms (eager) down to 8.06–8.08 ms (graphed), a 4.9× reduction with no algorithmic change, and 2.3× faster than softmax's 18.2–19.4 ms. Graphed interdomain holds ≈8.07 ms across all six measured prefix lengths from L=512 to L=16,384 (Figure 4). This regime is most relevant for single-request, interactive serving.

C.1.3. Throughput regime (B = 8–16)

At moderate batch sizes, launch overhead becomes a smaller but still substantial fraction of total decode time. Graphed interdomain stays prefix-invariant at 10.18 ms (B=8) and 12.64 ms (B=16), against softmax's $\sim$ 19 ms — a 1.5–1.9× latency advantage. This is also the regime where softmax first runs out of HBM: at B=16, L=16,384 softmax exhausts HBM3 during prefill, while graphed interdomain (with chunked prefill, see Section C.2) fits and continues to deliver 12.64 ms steady-state decode.

C.1.4. Compute-bound regime ($B \geq 32$)

Once batch size grows large enough for the per-step compute to dominate kernel-launch and KV-traffic costs, the graph-capture advantage shrinks. At B=32 graphed interdomain (17.34 ms) is only 1.10× faster than softmax (19.16 ms); at B=64 the ranking reverses: softmax (18.94–19.68 ms) is 30–36% faster than graphed interdomain (25.73–25.77 ms). The remaining advantage of interdomain in this regime is in memory rather than latency: at B=32 graphed interdomain still runs through L=8,192, where softmax has already exhausted HBM3 (max-fit L=4,096); and at B=64 graphed interdomain runs through L=4,096, where softmax stops at L=2,048.

Table 12. Peak VRAM (GB) during prefill at 1.3B on a single GH200 (120 GB HBM3), bfloat16 mixed precision (`torch.autocast`). "Softmax (SDPA)" and "Interdomain (non-chunked)" are the peak from a full-sequence prefill; "Chunked (C=2048)" is the peak from interdomain's chunked prefill, which retains only the running SSM state across chunks and discards per-chunk activations. "OOM" denotes out-of-memory.

B	L	Softmax (SDPA)	Interdomain (non-chunked)	Chunked (C=2048)
8	4,096	14.68	14.84	**12.75**
8	8,192	21.44	21.78	**13.25**
8	16,384	35.37	36.04	**14.63**
16	4,096	21.24	21.58	**17.30**
16	8,192	34.56	35.24	**18.07**
16	16,384	OOM	OOM	**19.99**
32	4,096	34.36	35.08	**26.42**
32	8,192	OOM	62.16	**27.74**

Table 13. Decode latency (ms/step, range over $L \in \{512, \ldots, 16{,}384\}$) at B=1 across the four LM scales. Speedup is the median softmax latency divided by the median graphed-interdomain latency. "Flatness" is the ratio of graphed steady-state latency at L=16,384 to that at L=512; values close to 1.0 indicate prefix-invariant decode.

Size	Softmax (ms)	Graphed (ms)	Speedup	Flatness
125 M	9.32–9.66	**2.92–2.93**	3.22×	1.00×
350 M	18.71–19.54	**6.05–6.06**	3.14×	1.00×
760 M	18.98–19.51	**6.95–7.01**	2.74×	1.01×
1.3 B	18.20–19.36	**8.06–8.08**	2.34×	1.00×

C.1.5. Limitations

The comparison is between graphed interdomain and eager softmax. A graphed softmax baseline would require a padded KV-cache allocation or a bucketed graph pool; this would close part of the launch-overhead gap at small B, but it would not eliminate the $\mathcal{O}(L)$ KV-cache memory traffic that becomes the bottleneck for softmax beyond the $L \leq 16{,}384$ window we measure here. Within that window, modern SDPA backends (cuDNN-FA on Hopper) amortise that traffic well enough that softmax decode is essentially prefix-invariant.

C.2. Prefill Memory Scaling

Chunked prefill processes the prompt in fixed-size chunks of C=2048 tokens, retaining only the running SSM state across chunks and releasing per-chunk activations. The peak memory is bounded by $\mathcal{O}(B \cdot C)$ instead of $\mathcal{O}(B \cdot L)$. At B=16, L=16,384, chunked interdomain uses 20.0 GB, versus 34.6 GB for softmax at L=8,192 (and OOM at L=16,384 on the same hardware). At B=32, L=8,192 even softmax OOMs: the $L \times L$ attention activations grow super-linearly in L, so the 34.4 GB peak at L=4,096 pushes past the 120 GB budget at L=8,192. Chunked interdomain uses only 27.7 GB and decodes through L=8,192 at ∼17.3 ms (Table 11).

C.3. Decode Latency Scaling Across Model Sizes

C.3.1. Reading

At B=1, graphed interdomain achieves a 2.3–3.2× latency advantage across all four scales, driven by elimination of per-step kernel-launch overhead. The advantage is largest at the smallest scale (3.22× at 125 M), where per-step compute occupies the smallest fraction of total step time, and shrinks monotonically with model size as per-step compute occupies a larger fraction of the total step time. At B=8 the same trend appears with smaller absolute speedups (2.61× → 1.86× from 125 M to 1.3 B), consistent with the compute-bound trend already visible in the $B \geq 32$ rows of Table 11. Across all eight cells of these two tables, graphed interdomain decode is prefix-invariant to within 1% from L=512 to L=16,384.

Table 14. Decode latency (ms/step) at $B{=}8$ across the four LM scales. The 1.3 B row entries match the $B{=}8$ row of Table 11.

Size	Softmax (ms)	Graphed (ms)	Speedup	Flatness
125 M	8.80–9.02	**3.37–3.40**	2.61×	1.01×
350 M	17.72–18.88	**7.21–7.30**	2.51×	1.01×
760 M	17.77–18.35	**8.51–8.54**	2.13×	1.00×
1.3 B	18.71–19.25	**10.16–10.20**	1.86×	1.00×