How Far Can Sliding Window Attention See?

Anonymous authors
Paper under double-blind review

Abstract

Sliding window attention (SWA) reduces the inference cost of Transformers by restricting each token to attend only to a fixed-size local window. However it remains unclear how information propagates beyond the attention window and, in turn, how long a sentence a SWA model can effectively manipulate. We provide an intuitive account of information propagation through successive SWA layers and use it to derive a theoretical prediction for the maximal effective length. For a model with window size W and ℓ layers, our analysis predicts a context length of WH_ℓ, where $H_\ell = \sum_{k=1}^{\ell} 1/k$ is the ℓ-th harmonic number. Experiments on the cramming task across several model families, scales, depths, and window sizes confirm this prediction, with predicted lengths within 20% of the measured limits in all tested configurations. Finally, we give an explicit construction that approximates full attention with SWA.

1 Introduction

Transformers (Vaswani et al., 2017) have become the dominant architecture in modern deep learning, achieving state-of-the-art performance across language and vision (Wolf et al., 2020; Dosovitskiy et al., 2021). One of their main limitations is inference time, which scales quadratically with sequence length. Sliding window attention (SWA) offers a simple and powerful alternative (Beltagy et al., 2020; GemmaTeam et al., 2025), reducing attention complexity to linear by restricting each token to a local context. This context is called the attention window, and we denote its size by W. Beyond its intuitive appeal, this design has seen widespread practical adoption in modern efficient architectures. Models such as Mistral (Jiang et al., 2023), Gemma 3 (GemmaTeam et al., 2025), and GPT-OSS (OpenAI et al., 2025) use sliding window layers as a core component, often combined with global attention, while inference methods such as StreamingLLM (Xiao et al., 2024b) and DuoAttention (Xiao et al., 2024a) similarly pair a bounded sliding window with sparse mechanisms for long-range access.

However, this efficiency comes at the cost of context length scaling. While the latter has been studied empirically (Assogba & Ren, 2025; Willette et al., 2025; Nawrot et al., 2026), there are little to no theoretical studies. We aim to bridge this gap. There is one clear limitation to SWA: as each layer can propagate information by at most W positions, a sliding window transformer with ℓ layers cannot handle contexts longer than $W\ell$. Our purpose in this paper will be to show how information loss throughout SWA leads to an effective maximal context length of WH_ℓ where $H_\ell = \sum_{k=1}^{\ell} 1/k$ is the ℓ-harmonic number. Our paper is structured as follows.

- We introduce the notations and provide the definition of SWA in Section 2.
- We provide an intuitive account of information propagation through successive SWA layers in Section 3.
- We show that SWA can approximate full attention to arbitrary precision in Section 4. We also show how the WH_ℓ formula can be guessed from studying the optimality of our construction.
- We discuss the necessary assumptions and prove formally the WH_ℓ formula in Section 5.
- Finally, we show empirically the validity of our theory in Section 6.

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

1.1 Related Work

While SWA has been extensively studied empirically, theoretical work remains limited and has mostly focused on formal language theory.

Empirical studies of Sliding Window Attention. A consistent finding across benchmarks is that SWA degrades as context grows beyond the window. Assogba & Ren (2025) show that Mistral-7B-v0.1 with SWA struggles to retrieve keys located more than one attention window away. Willette et al. (2025) observe the same degradation at scale, tracking the drop in SWA performance as context length increases on the 131K RULER benchmark. Subsequent work situates SWA within a broader design space: Nawrot et al. (2026) compare how the performance of different sparse-attention strategies scales with context length and sparsity, while Cabannes et al. (2026) conduct related experiments for hybrid architectures that combine SWA with recurrent components. Together, these studies establish that SWA degrades with context length, but they do not identify the responsible mechanism.

Theoretical studies of Sliding Window Attention. Theoretical work, by contrast, has emphasized what SWA can express rather than how it behaves at long context. Yun et al. (2020) establish that the universal approximation theorem still holds for sparse attention. Interestingly, SWA has proven to improve the expressivity of transformers from a formal language standpoint. Li et al. (2026) show that SWA allows transformers to remain Turing complete even without positional encoding. Li & Cotterell (2026) sharpen this picture by characterizing the languages sparse-attention transformers can recognize. In contrast, we study a problem closer to practical deployment: what is the maximal context length of SWA?

The Cramming Task We study this question through the lens of the cramming task (Kuratov et al., 2025). This task measures the maximal sentence length a transformer can reliably output, providing an effective way to both formalize our problem and test it empirically. The work most closely related to ours is that of Meyer et al. (2026), who use this task to characterize the expressivity limits of regular transformers. While their theoretical upper bounds still apply to sliding window transformers, in this work, we derive tighter bounds that capture the inherent limitations of SWA.

1.2 Notations

We write $[n] = \{0, \dots, n\}$ for all $n \in \mathbb{N}$ (we take $\mathbb{N}$ to be the set of strictly positive integers). Bold lowercase letters (e.g., $\mathbf{x}$) denote vectors, and bold uppercase letters (e.g., $\mathbf{W}$) denote matrices. For a matrix $\mathbf{W}$, we write $\mathbf{W}_{i,j}$, $\mathbf{W}_{i,:}$, and $\mathbf{W}_{:,j}$ to refer to its (i,j)-th element, i-th row, and j-th column, respectively. We can use $i = -1$ and $j = -1$ to denote the last row and column. The concatenation of k vectors $\mathbf{x}^1 \in \mathbb{R}^{d_1}, \dots, \mathbf{x}^k \in \mathbb{R}^{d_k}$ is denoted as $[\mathbf{x}^1, \dots, \mathbf{x}^k] \in \mathbb{R}^{\sum_{i=1}^k d_i}$. We extend the standard infinity norm $\|\mathbf{x}\|_\infty = \max_i |\mathbf{x}_i|$ to sequences as $\|(\mathbf{x}_1, \dots, \mathbf{x}_n)\|_\infty = \max_{t\in[n]} \|\mathbf{x}_t\|_\infty$. We denote by $B^d(0, r) = \{\mathbf{x} \in \mathbb{R}^d, \|\mathbf{x}\|_\infty \leq r\}$ the d-dimensional ball of radius r.

2 The Transformer Architecture

2.1 Standard Attention

Definition 1 (Transformer layer)**.** A single-head decoder transformer layer is defined as a map

$$\mathrm{L} : \bigcup_{n=1}^{+\infty} \mathbb{R}^{d\times n} \to \bigcup_{n=1}^{+\infty} \mathbb{R}^{d\times n} \quad \text{that maps} \quad (\mathbf{h}_1, \dots, \mathbf{h}_n) \mapsto (\mathbf{h}_1^+, \dots, \mathbf{h}_n^+),$$

which consists of a composition of a multilayer perceptron MLP that sends $\mathbf{h}_t^\sim$ to $\mathbf{h}_t^+$(Definition 3) with an attention component Att that sends $\mathbf{h}_t$ to $\mathbf{h}_t^\sim$ (Definition 2):

$$\mathbf{h}_t^+ = \mathbf{h}_t^\sim + \mathrm{MLP}(\mathbf{h}_t^\sim), \qquad \mathbf{h}_t^\sim = \mathbf{h}_t + \mathrm{Att}(\mathbf{h}_1, \dots, \mathbf{h}_t).$$

Definition 2 (Attention component)**.** Fix query, key and value matrices $\mathbf{Q}, \mathbf{K} \in \mathbb{R}^{s\times d}, \mathbf{V} \in \mathbb{R}^{d\times d}$, and write $\mathbf{q}_t = \mathbf{Q}\mathbf{h}_t, \mathbf{k}_i = \mathbf{K}\mathbf{h}_i$. Then an attention head is defined by $\mathrm{Att}^i(\mathbf{h}_1, \dots, \mathbf{h}_t) := \mathbf{V}^i \sum_{j=1}^t \alpha_{tj}^i \mathbf{h}_j^i$, with $\alpha_{tj}^i = \frac{\exp(\mathbf{q_t^i}^\top \mathbf{k}_j^i)}{\sum_{m=1}^t \exp(\mathbf{q_t^i}^\top \mathbf{k}_m^i)}$. General attention consists in a sum of attention heads $\mathrm{Att}(\mathbf{h}_1, \dots, \mathbf{h}_t) = \sum_{i=1}^{n_{\text{heads}}} \mathrm{Att}^i(\mathbf{h}_1, \dots, \mathbf{h}_t)$.

Definition 3 (Multilayer perceptron)**.** A multilayer perceptron with length $L \geq 2$ and width $m \geq 1$ is a map $\mathrm{MLP} : \mathbb{R}^d \to \mathbb{R}^d$ of the form

$$\mathrm{MLP}(\mathbf{x}) = \mathbf{W}_L \sigma(\mathbf{W}_{L-1}\sigma(\cdots \sigma(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1)\cdots) + \mathbf{b}_{L-1}) + \mathbf{b}_L,$$

where $\sigma : \mathbb{R} \to \mathbb{R}$ is an activation function applied coordinatewise,

$$\mathbf{W}_1 \in \mathbb{R}^{m\times d}, \qquad \mathbf{W}_l \in \mathbb{R}^{m\times m} \text{ for } 2 \leq l \leq L-1, \qquad \mathbf{W}_L \in \mathbb{R}^{d\times m},$$
$$\text{and} \qquad \mathbf{b}_l \in \mathbb{R}^m \text{ for } 1 \leq l \leq L-1, \qquad \mathbf{b}_L \in \mathbb{R}^d.$$

Remark 4. The definitions here omit the layer normalization block for simplicity, as is standard in theory works on transformers Wei et al. (2022); Wang et al. (2023b); Jiang & Li (2024); Kajitsuka & Sato (2025); Furuya et al. (2025); Yang et al. (2026).

2.2 Sliding Window Attention

In sliding window attention with window size $W \geq 1$, every token $\mathbf{h}_t$ can only attend to the W previous ones. That is

$$\mathbf{h}_t^{\sim} = \mathbf{h}_t + \mathrm{Att}(\mathbf{h}_{\max(1,t-W)}, \ldots, \mathbf{h}_t).$$

A transformer where full attention is replaced with sliding window attention is called a sliding window transformer.

3 An Intuitive View of Information Propagation in Sliding Window Transformers

A single SWA layer cannot attend beyond its local window, but stacking layers allows distant tokens to influence a prediction indirectly. We provide a visual representation of how this information propagation takes place in practice using activation patching (Geiger et al., 2021; Wang et al., 2023a; Zhang & Nanda, 2024).

Concretely, we consider sentences where the next token $\mathbf{t}_{\text{next}}$ depends entirely on a previous token $\mathbf{t}_{\text{prev}}$. If $\mathbf{t}_{\text{prev}}$ lies within the information window, its information can reach the final token directly, allowing the model to predict $\mathbf{t}_{\text{next}}$. When $\mathbf{t}_{\text{prev}}$ lies several windows away, however, its information must instead be passed across windows through a sequence of relay tokens.

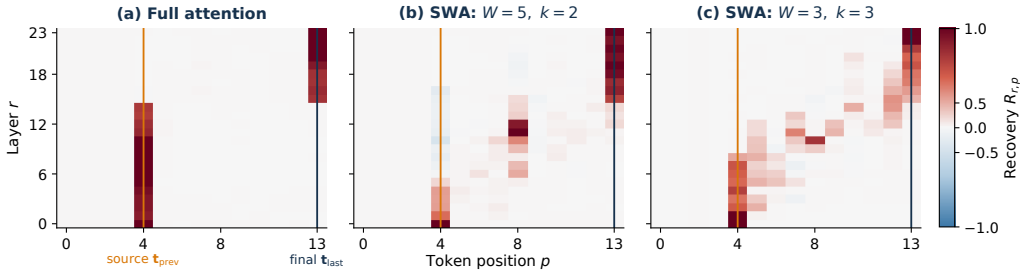


Figure 1: **Information propagation in SWA versus full attention on Qwen2.5-0.5B.** We run Qwen2.5-0.5B on the prompt pair "The traffic light was `green/red`. The driver approached the intersection and decided to …". At each layer r and position p, we plot how much of the "traffic color" information is contained in hidden state $\mathbf{h}_{r,p}$. The orange and navy lines mark $\mathbf{t}_{\text{prev}}$ and $\mathbf{t}_{\text{last}} = \texttt{to}$. Full attention requires no relay token, while SWA of size $W = 5$ requires one at position 8, and more are added for $W = 3$.

To visualize where the information is stored, we compare two sentences where only $\mathbf{t}_{\text{prev}}$ differs (e.g. "I am a `boy/girl`. My gender is …", $\mathbf{t}^1_{\text{prev}} = \texttt{boy}, t^2_{\text{prev}} = \texttt{girl}, \mathbf{t}^1_{\text{next}} = \texttt{male}, \mathbf{t}^2_{\text{next}} = \texttt{female}$). Write

- $z^{i\to j}$ the next-token logit of $\mathbf{t}^j_{\text{next}}$ when $\mathbf{t}_{\text{prev}} = \mathbf{t}^i_{\text{prev}}$ (since $\mathbf{t}^i_{\text{next}}$ is the most logical continuation to $\mathbf{t}^i_{\text{prev}}$, $z^{i\to i}$ should be high while $z^{i\to j}$ should be low for $i \neq j$).

- $m^i = z^{i\to 1} - z^{i\to 2}$ is the directional output margin. Positive values favour $\mathbf{t}^1_{\text{next}}$, while negative values favour $\mathbf{t}^2_{\text{next}}$. The magnitude measures the strength of this pairwise preference.
- $\mathbf{h}^i_{r,p}$ the hidden state corresponding to layer r and position p when $\mathbf{t}_{\text{prev}} = \mathbf{t}^i_{\text{prev}}$.

We take the input to be $\mathbf{t}^1_{\text{prev}}$ and replace one of the $\mathbf{h}^1_{r,p}$ by the corresponding $\mathbf{h}^2_{r,p}$. If this diminishes the next-token logit of $\mathbf{t}^1_{\text{next}}$ and increases that of $\mathbf{t}^2_{\text{next}}$, then it means that the information was stored in $\mathbf{h}_{r,p}$. Formally, if $m_{r,p}$ is the margin after modifying the hidden state (r, p), we plot in Figures 1 and 2 the normalized recovery $R_{r,p} = \frac{m^1 - m_{r,p}}{m^1 - m^2}$. Clearly, $R_{0,0} = 1$ as this amounts to replacing $\mathbf{t}^1_{\text{prev}}$ by $\mathbf{t}^2_{\text{prev}}$ (we write 0 the position of $\mathbf{t}^i_{\text{prev}}$). Simmilarly, $R_{l,d} = 1$ as this amounts to replacing the representation of the last token at the last layer, which entirely determines $\mathbf{t}_{\text{next}}$ (we write d the position of the last token). Moreover, if p denotes the position of a relay token which stores the information at layer r, then $R_{r,p}$ should be closer to 1. Conversely, $R_{r,p}$ should be closer to 0 if the token p stores no information at layer r. In particular, this means that $R_{l,p} = 0$ for $p \neq d$, as only the last token matters at the last layer. We illustrate this in Figure 2.

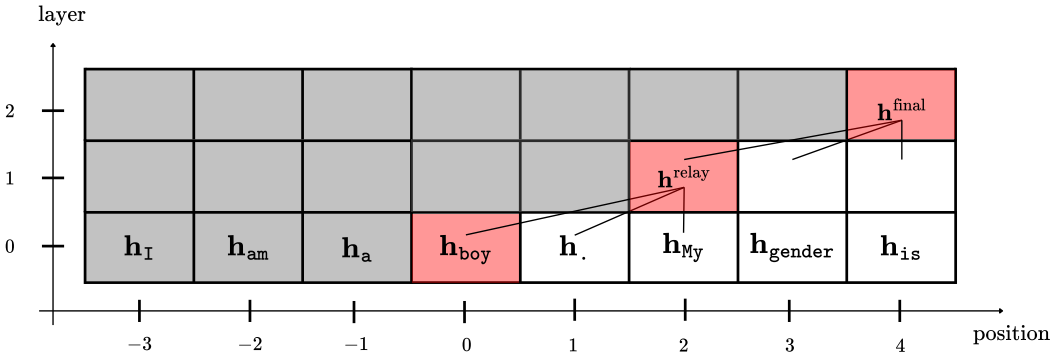


Figure 2: **Illustration of the heatmap.** The input is "I am a boy. My gender is". In the first layer, the corresponding hidden states are shown. The next token is entirely predicted by the final hidden state at the last layer $\mathbf{h}^{\text{final}}$. The sliding window is of size $W = 2$ and there are $l = 2$ layers. The grey hidden states therefore have no impact on the prediction and correspond to $R = 0$. Since boy is two windows away, the *male* information is stored in the antepenultimate token at the first layer, before being transfered to $\mathbf{h}^{\text{final}}$ at the second layer. Clearly, $R_{0,0} = R_{2,4} = 1$. The closer $R_{1,2}$ is to 1, the more $\mathbf{h}^{\text{relay}}$ acts as a relay (with it storing the full information if $R_{1,2} = 1$).

Figure 1 compares full attention with SWA for two window sizes. In both SWA cases, the `green` prompt ranks `go` above `stop`, while the `red` prompt ranks `stop` above `go`. Exact experimental details are given in Appendix A.

Through the lens of the WH_ℓ formula. This section shows that under SWA, each token carries two kinds of information: its own content, and that of the previous tokens outside the current window. This should lead to a bottleneck. Indeed, a hidden state needs to store the information of an increasing number of tokens as the sentence grows in length. Figure 1 shows this effect. As the window narrows, relayed information spreads across more tokens rather than concentrating in a few, and no token acts as a clear relay.

We further formalize this phenomenon in the following sections.

4 Every Transformer Can be Approximated by a Sliding Window Transformer

We show in this section that any transformer can be approximated by a sliding window transformer of window size 1, and hence of arbitrary window size.

We keep the notation of Section 2: $\mathrm{L}(\mathbf{h}_1, \ldots, \mathbf{h}_n) = (\mathbf{h}^+_1, \ldots, \mathbf{h}^+_n)$.

As sliding window attention is inherently less expressive than its regular counterpart, it is necessary to increase the hidden dimension for the approximation to be possible. We therefore extend every input and output $\mathbf{h}_t, \mathbf{h}_t^+ \in \mathbb{R}^d$ to $\mathbb{R}^{2Nd+1}$ using the following injection:

$$\overline{\mathbf{h}_t} = [\mathbf{h}_t, \mathbf{0}, \ldots, \mathbf{0}, t], \qquad \overline{\mathbf{h}_t^+} = [\mathbf{h}_t^+, \mathbf{0}, \ldots, \mathbf{0}, t].$$

Theorem 5 (Approximation of a regular transformer layer by a sliding window transformer of window size 1)**.** *Fix $N \geq 1$, $r > 0$, and $\varepsilon > 0$. Let L be a single-head decoder attention layer as defined in Definition 1.*

Then there exists a N-layer sliding-window attention $\widetilde{\mathrm{L}}_N$ of window size 1, hidden dimension $2Nd+1$ and MLP size $f(\varepsilon)$ (discussed in Remark 7) that approximates L up to error ε on sentences of length at most N.

That is for every $n \leq N$ and every $\mathbf{h}_1, \ldots, \mathbf{h}_n \in B^d(0, r)$ we have

$$\|\widetilde{\mathrm{L}}_N(\mathbf{0}, \overline{\mathbf{h}_1}, \ldots, \overline{\mathbf{h}_n}) - (\mathbf{0}, \overline{\mathbf{h}_1^+}, \ldots, \overline{\mathbf{h}_n^+})\|_\infty \leq \varepsilon.$$

Remark 6. The assumption $\mathbf{h}_1, \ldots, \mathbf{h}_n \in B^d(0, r)$ for some $r > 0$ independent of the $\mathbf{h}_i$ holds in practice because of normalization. This can be proved both empirically and theoretically (Meyer et al., 2026).

Remark 7. The required size $f(\varepsilon)$ of the MLP depends on its parameters, such as the activation functions being used and any constraints on the MLP activation function, depth, and width. Classical approximation results make this dependence explicit once these choices are fixed; for example, Lu et al. (2021) provides simultaneous depth–width bounds for ReLU networks.

Sketch of Proof. The main difficulty is that, in sliding-window attention with window size 1, each position can only attend to itself and to the immediately preceding position. Hence token t cannot directly access the whole prefix $(\mathbf{h}_1, \ldots, \mathbf{h}_t)$, unlike in standard attention. We therefore introduce $n-1$ shift blocks. Their role is to progressively move information one step to the right, so that after enough shifts, position t has collected $(\mathbf{h}_t, \mathbf{h}_{t-1}, \ldots, \mathbf{h}_1)$. Once this prefix is stored locally at position t, a final computation block can reproduce the standard attention update and output $\mathbf{h}_t^+$. Thus one full attention layer is approximated by $n-1$ shift blocks followed by one computation block, as shown in Figure 3. □

The full construction can be found in Appendix B.

As stated above, SWA is inherently less expressive than its regular counterpart. It is thus necessary to increase the model size in order to reach the same expressivity. We increase it in three ways:

- We multiply the number of layers by a factor N. Using a larger window size W allows us to obtain a $\Theta(\frac{N}{W})$ scaling, as shown in Appendix C.2. This scaling is optimal, as shown in Appendix D.
- We increase the hidden dimension from d to $2Nd+1$. This can be further improved on, as discussed in Section 4.1.
- The only other characteristics we increase are MLP width and length. However, Meyer et al. (2026) suggest that we can disregard them for the cramming task analysis. A more in depth discussion is provided in Appendix E.

4.1 Improving the Hidden Dimension Scaling

The construction in Theorem 5 stores the whole prefix at position t. This uses N d-dimensional slots in the main state. However, this is not necessary. For window size 1, position t can read position $t-1$ in the final computation block. Hence position t only needs to store tokens with the same parity as t, while position $t-1$ stores the tokens with the opposite parity. This reduces the number of slots by a factor 2. We provide the detailed construction in Appendix C.1, and generalize to arbitrary window size in Appendix C.2. And we show in Appendix F that RoPE allows to improve the hidden dimension to $O(\frac{Nd}{W})$.

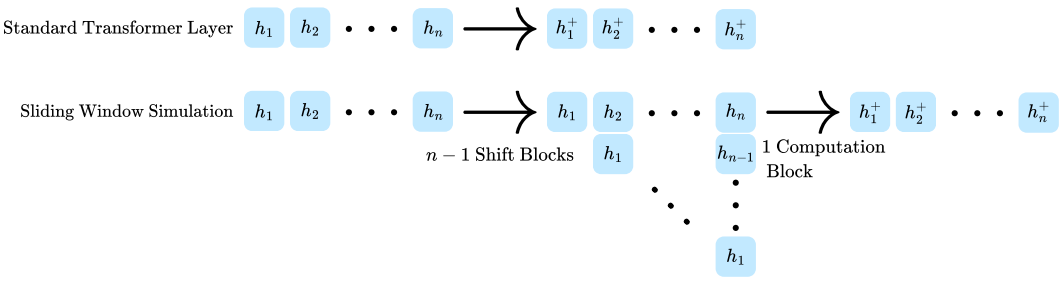


Figure 3: Approximation of one standard transformer layer by a sliding-window transformer with $W = 1$.

4.2 Through the Lens of the WH_ℓ Formula

As briefly discussed in Section 3, the loss of information comes from the fact that in SWA, tokens are burdened to transfer information from previous windows. Intuitively, the more information gets compressed into one window, the higher the information loss. A natural way to quantify this information loss would be the following:

If a token needs to store the information from w previous tokens on top of his own, then the information gets diluted by a factor $\frac{1}{w+1}$.

This is backed up by the proofs of Theorem 5, provided the construction in Appendix F is optimal. Indeed, let us assume that the optimal approximation of full attention with SWA requires a hidden dimension equal to id in the i^{th} window (as in Figure 3, where we see the hidden dimension increase in the sliding window simulation). If we take $N = W\ell$ to be the maximal sequence length, then the hidden dimension of the i^{th} token is id for $i \in [W\ell]$. The extension of Section 4.1 shows that we can group tokens by windows and have the hidden dimension be respectively of order $\{d, \dots, \ell d\}$ for the ℓ attention windows. We would therefore get an inverse factor when recovering the expressive power of SWA from full attention.

Therefore, if considering the information loss makes us go from ℓ windows working at capacity 1 (in the ideal setting where the $W\ell$ horizon is reached), to windows working only at capacity $1, \frac{1}{2}, \dots, \frac{1}{\ell}$ respectively, then the effective horizon should be given by

$$\text{full horizon} \times \frac{1 + \frac{1}{2} + \ldots + \frac{1}{\ell}}{1 + 1 + \ldots + 1} = W\ell \times \frac{H_\ell}{\ell} = WH_\ell.$$

We formalize this intuition in the following section.

5 Formalizing the Intuition

In order to prove our main result, we require a formal definition of what it means for SWA "to see up to a given distance". To that end, we introduce the cramming task in Section 5.1. We then discuss the necessary assumptions and prove the WH_ℓ formula in Section 5.2.

5.1 The Cramming Task

In this section, we leverage the cramming task (Kuratov et al., 2025) to formally define what it means for SWA "to see up to a given distance".

Definition 8 (Cramming)**.** For any target sequence $[\mathbf{x}_1, \dots, \mathbf{x}_c]$, a set of trainable memory vectors $[\mathbf{mem}] = [\mathbf{y}_1, \dots, \mathbf{y}_m]$ is optimized so that the model generates $[\mathbf{x}_1, \dots, \mathbf{x}_c]$ when prompted with $[\mathbf{mem}]$. This approach effectively searches over all possible prompts—including highly unnatural ones—to determine whether a sequence is accessible.

This provides the perfect framework to study the maximal sentence length SWA can handle: we define it as the cramming horizon.

Definition 9 (Cramming horizon)**.** The Cramming Horizon C of a model $\mathcal{M}$ is the maximal sentence length that it can reliably generate. More formally, write

- $\mathcal{I}$ the input set of $\mathcal{M}$ (as discussed in Remark 6, we can take $\mathcal{I} = B^d(0, r)$),
- $\mathcal{V}$ its vocabulary (that is the set of all possible tokens),
- $\mathcal{A}_c^{\mathcal{M}} = \{\mathbf{X} \in \mathcal{V}^c, \exists \mathbf{Y} \in \bigcup_{n\in\mathbb{N}} \mathcal{I}^n\}$ the set of accessible sentences of length c (that is those that $\mathcal{M}$ can generate for some soft prompt),
- $a_c^{\mathcal{M}} = \frac{|\mathcal{A}_c^{\mathcal{M}}|}{|\mathcal{V}|^c}$ the proportion of accessible sentences.

Then $C = \max\{c \in \mathbb{N}, a_c^{\mathcal{M}} \geq 50\%\}$. We chose 50% here as a cutoff. This arbitrary choice is harmless as Meyer et al. (2026) show both theoretically and empirically that $a_c^{\mathcal{M}}$ decreases exponentially fast for standard attention. We verify empirically this also holds for SWA in Appendix J.5.

We can now formally state our problem:

What is the maximal cramming horizon of Sliding Window Attention?

5.2 Proving the WH_ℓ Formula

We show in Section 5.2.1 that the $W\ell$ limit is tight when information is preserved. However, Section 3 suggests that information degrades in practice. This motivates Section 5.2.2, where we show that such information loss yields the effective-context scaling WH_ℓ.

5.2.1 The clear $W\ell$ upper bound

As mentioned in Section 3, a clear upper bound on the cramming horizon is $W\ell$. We prove this formally in Appendix D. Moreover, this bound is tight.

Theorem 10 (Tight cramming horizon)**.** *For every $W, \ell \geq 1$, there is an ℓ-layer causal sliding-window transformer of window size W whose output at every position $W\ell < t \leq N$ is the input token at position $t - W\ell$. Consequently, the maximum cramming horizon of an ℓ-layer SWA transformer is exactly $W\ell$.*

We provide the construction in Appendix G. Note that the SWA construction used is highly unnatural: every predicted token depends only on the token located W positions before. However, as shown in Section 3, the limitation of SWA in practice comes precisely from the fact that information gets lost when trying to merge the information from several previous tokens in order to predict the next one. We therefore need a more precise theoretical framework to map what is happening in practice.

5.2.2 Information-Theoretic Effective Context

The previous section motivates distinguishing the theoretical cramming horizon from an *effective* cramming horizon, which does not rely on the assumption that information is perfectly preserved during propagation. Although a transformer is deterministic, the representation of a fixed source token varies with the remaining context. We therefore treat the source as fixed and the surrounding context as nuisance randomness. Conditional on the attention weights, one attention update can be decomposed as

$$\alpha_s \mathbf{V}\mathbf{h}_s + \sum_{i\neq s} \alpha_i \mathbf{V}\mathbf{h}_i,$$

where the first term is the source contribution and the second represents interference from the remaining context. Equivalently,

$$\mathbf{h}^{j+1} = \mathbf{A}_j \mathbf{h}^j + \boldsymbol{\varepsilon}_j, \tag{1}$$

Similar modelisations have been used to analyze various architectures. For instance, dynamical mean-field analyses (Sompolinsky et al., 1988) approximate recurrent inputs by Gaussian

noise (Mastrogiuseppe & Ostojic, 2017; 2019; Qin & Sharpee, 2026). Schröder et al. (2024) derive Gaussian-equivalent descriptions of feature maps in high-dimensional regimes, and Goldfeld et al. (2019) models information propagation through neural networks as additive white Gaussian noise channels.

We use Fisher information, defined in Appendix H, to quantify how accurately the intended value $\alpha_s \mathbf{V}\mathbf{h}_s$ can be recovered from the final received representation after k window crossings. Under the assumption that the Fisher information is proportional to the cramming horizon and with Gaussian noise, we can prove the following.

Theorem 11 (Informal)**.** *The Cramming horizon of an ℓ layers SWA transformer with window size W is given by the formula*

$$\boxed{C = W H_\ell}.$$

We provide the details in Appendix H.

6 Experiments

Theorem 11 predicts that an ℓ-layer sliding-window transformer with window size W has a cramming horizon of $W H_\ell$ tokens. We now test this prediction. Section 6.1 describes how we implement the cramming task of Section 5.1. We then discuss how to identify the cramming horizon in Section 6.2. Finally, Section 6.3 measures it on eight models at three window sizes and compares it with the predicted $W H_\ell$.

6.1 Implementing the cramming task

We use the cramming task introduced by Kuratov et al. (2025) and follow the experimental setup of Meyer et al. (2026). The task tests whether a frozen transformer can reproduce a target sequence $\mathbf{x}$ by optimizing only a soft prompt $\mathbf{Y}$. In practice, $\mathbf{x}$ is sampled uniformly at random from a list of 100,000 words. Because the models use different tokenizers, the same words may produce different numbers of tokens. We use T to denote the number of tokens in the resulting target sequence. We fix $M \in \mathbb{N}$ and obtain $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_M)$ by minimizing the cross-entropy loss $\mathcal{L}(\mathbf{Y}; \mathbf{x}_{1:T}) = -\sum_{i=1}^{T} \log p_\theta(\mathbf{x}_i \mid [\mathbf{Y}, \mathbf{x}_{1:i-1}])$.

Let $\mathcal{V}$ denote the vocabulary of the model (the set of all possible tokens). A trial succeeds only if every target token is predicted correctly: $\arg\max_{t \in \mathcal{V}} p_\theta(t \mid [\mathbf{Y}, \mathbf{x}_{1:i-1}]) = \mathbf{x}_i$ for every $i \in [T]$.

For fixed M, W, and ℓ, let $\widehat{p}_{M,W,\ell}(T)$ denote the fraction of sampled length-T targets recovered exactly after optimization. Following Meyer et al. (2026), we report the 50% threshold

$$\widehat{T}_{\leq 50}(M, W, \ell) = \min\{T : \widehat{p}_{M,W,\ell}(T) \leq 0.5\}.$$

6.2 SWA saturates at the predicted cramming horizon

Clearly, a longer memory size M allows to encode more information, and therefore to access longer sequence lengths T. While Meyer et al. (2026) established a theoretical limitation on the maximal accessible sequence length T for standard transformers, it is in practice extremely high and has never been verified. However, our theory predicts that SWA should have a much smaller limit of $H_\ell W$.

We test this on Qwen2.5-0.5B (which has 24 decoder layers) for various window sizes $W \in \{16, 32, 64\}$. For each window, we vary the target length T in 10-token increments and conduct 20 independent trials for each T. We report $\widehat{T}_{\leq 50}$ as the first length at which the exact-recovery rate falls to 50% or below. We do this for increasing memory size $M \in \{1, 2, 3, 4, 5, 10, 50, 100, 500, 1000\}$.

Figure 4 shows two clear regimes. For small memory size, $\widehat{T}_{\leq 50}$ increases similar to regular transformers. But once M is sufficiently large, adding memory no longer increases $\widehat{T}_{\leq 50}$, and the curves flatten near 50, 140, and 220 tokens respectively for $W = 16, 32, 64$. The corresponding harmonic references are $H_{24}W = 60.4,\ 120.8,\ 241.6$. The same saturating shape appears on Qwen3-1.7B in Appendix J.1. We refer to this plateauing value as the cramming horizon.

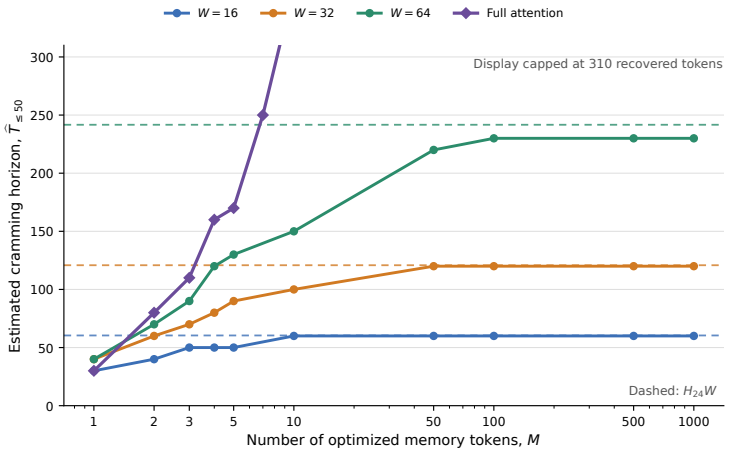


Figure 4: **The cramming horizon of Qwen2.5-0.5B SWA saturates at the predicted $H_{24}W$.** The maximal accessible length $\widehat{T}_{\leq 50}$ first grows with the memory size M, then plateaus (solid curves). Each plateau lands on the theoretical prediction $H_{24}W$ (dashed lines). Full attention (purple) shows no such plateau and leaves the display range.

6.3 The prediction holds across models and window sizes

We now test the predicted scale WH_ℓ. We fix $M = 200$, well past the plateau of Section 6.2, and estimate the cramming horizon for $W \in \{16, 32, 64\}$ on eight models with 24 to 64 layers, listed in Appendix J.3. A 20-trial sweep over target lengths first locates the 50% crossing. We then run 80 further trials at each of the two 10-token lengths that bracket it, so the reported crossing rests on 100 trials per side. Weights stay frozen and the memory vectors are optimized for at most 5,000 steps.

Table 1 reports the ratio $\widehat{T}_{\leq 50}/(WH_\ell)$. Two facts stand out. First, within every model the ratio is nearly constant across W, confirming the linear scaling. Second, the ratio lies in $[0.79, 1.16]$. This validates our theoretical prediction WH_ℓ. We conduct further experiments to isolate the effect of the number of layers in Appendix J.4.

Table 1: **The measured cramming horizon matches the predicted WH_ℓ across every model and window size.** Each window cell reports $\widehat{T}_{\leq 50}/(WH_\ell)$ followed by their ratio in brackets, using the model's listed layer count to compute H_ℓ. A dagger marks an earlier 20-trial estimate; all other crossings have 100 trials at both adjacent 10-token endpoints.

Model	Parameters	Layers	$W = 16$	$W = 32$	$W = 64$
Qwen2.5-0.5B†	0.5B	24	50/60 [0.83]	140/121 [1.16]	220/242 [0.91]
Qwen3-0.6B	0.6B	28	70/63 [1.11]	140/126 [1.11]	270/251 [1.07]
Qwen3-1.7B†	1.7B	28	60/63 [0.95]	140/126 [1.11]	290/251 [1.15]
Qwen3-4B	4.0B	36	70/67 [1.05]	130/134 [0.97]	270/267 [1.01]
Qwen3-8B†	8.0B	36	70/67 [1.05]	130/134 [0.97]	270/267 [1.01]
Qwen3-32B	32B	64	70/76 [0.92]	120/152 [0.79]	240/304 [0.79]
Gemma-2-2B	2.6B	26	70/62 [1.14]	130/123 [1.05]	260/247 [1.05]
Gemma-2-9B	9.2B	42	70/69 [1.01]	130/138 [0.94]	240/277 [0.87]

7 Conclusion

Experiments on the cramming task show that there exists a model-dependent threshold C_{model} on the effective length SWA can handle: while a fixed model can generate almost any sentence of length less than C_{model}, it can generate almost no sentence of length greater than C_{model}. We provided intuition, proved theoretically, and validated empirically the formula $C_{\text{model}} = WH_\ell$.

AI USE STATEMENT

In this work, we used generative AI tools for checking typos, writing proofs in latex from handwritten notes, helping with writing code, and discussing hypotheses (in particular finding relevant previous work doing a particular hypothesis). We have not used generative AI tools to generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, interpret results. Additionally, we used generative AI tools to create or modify scientific figures or images, create or edit software code, summarize or analyse existing literature, brainstorming, sourcing/searching for information, identify relevant literature. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Yannick Assogba and Donghao Ren. Evaluating long range dependency handling in code generation LLMs. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=yzACI2vFaX.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020. URL https://arxiv.org/abs/2004.05150.

Loïc Cabannes, Maximilian Beck, Maria Lomeli, Gergely Szilvasy, Matthijs Douze, Jade Copet, Pierre-Emmanuel Mazare, Gabriel Synnaeve, and Herve Jegou. Short window attention enables long-term memorization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=btgVfhudI1.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Takashi Furuya, Maarten V. de Hoop, and Gabriel Peyré. Transformers are universal in-context learners. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=6S4WQD1LZR.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/4f5c422f4d49a5a807eda27434231040-Abstract.html.

GemmaTeam et al. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Ziv Goldfeld, Ewout Van Den Berg, Kristjan Greenewald, Igor Melnyk, Nam Nguyen, Brian Kingsbury, and Yury Polyanskiy. Estimating information flow in deep neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2299–2308. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/goldfeld19a.html.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Haotian Jiang and Qianxiao Li. Approximation rate of the transformer architecture for sequence modeling. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=ZwS2y21mZV.

Tokio Kajitsuka and Issei Sato. On the optimal memorization capacity of transformers. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=UGVYezlLcZ.

Yuri Kuratov, Mikhail Arkhipov, Aydar Bulatov, and Mikhail Burtsev. Cramming 1568 tokens into a single vector and back again: Exploring the limits of embedding space capacity. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19323–19339, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.948. URL https://aclanthology.org/2025.acl-long.948/.

Jiaoda Li and Ryan Cotterell. Characterizing the expressivity of local attention in transformers. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 37485–37507, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1739. URL https://aclanthology.org/2026.acl-long.1739/.

Qian Li, Xinyu Mao, and Shang-Hua Teng. Rethinking the role of positional encoding: Sliding-window transformers without pe remain turing complete, 2026. URL https://arxiv.org/abs/2606.01532.

Jianfeng Lu, Zuowei Shen, Haizhao Yang, and Shijun Zhang. Deep network approximation for smooth functions. *SIAM Journal on Mathematical Analysis*, 53(5):5465–5506, 2021. doi: 10.1137/20M134695X. URL https://doi.org/10.1137/20M134695X.

Francesca Mastrogiuseppe and Srdjan Ostojic. Intrinsically-generated fluctuating activity in excitatory-inhibitory networks. *PLOS Computational Biology*, 13(4):1–40, 04 2017. doi: 10.1371/journal.pcbi.1005498. URL https://doi.org/10.1371/journal.pcbi.1005498.

Francesca Mastrogiuseppe and Srdjan Ostojic. A geometrical analysis of global stability in trained feedback networks. *Neural computation*, 31(6):1139–1182, 2019.

Maxime Meyer, Mario Michelessa, Caroline Chaux, and Vincent Y. F. Tan. How many different outputs can a transformer generate? In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=x4xOtzoYa6.

Piotr Nawrot, Jianing Li, Renjie Huang, Sebastian Ruder, Kelly Marchisio, and Edoardo Ponti. The sparse frontier: Sparse attention trade-offs in transformer LLMs. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 38667–38701, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1926. URL https://aclanthology.org/2026.findings-acl.1926/.

OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, et al. gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

Haodong Qin and Tatyana Sharpee. Information dynamics and memory in neural networks through fisher information diffusion. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=cvfjjRCY0U.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Dominik Schröder, Daniil Dmitriev, Hugo Cui, and Bruno Loureiro. Asymptotics of learning with deep structured (Random) features. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 43862–43894. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/schroder24a.html.

H. Sompolinsky, A. Crisanti, and H. J. Sommers. Chaos in random neural networks. *Phys. Rev. Lett.*, 61:259–262, Jul 1988. doi: 10.1103/PhysRevLett.61.259. URL https://link.aps.org/doi/10.1103/PhysRevLett.61.259.

A.J. Stam. Some inequalities satisfied by the quantities of information of fisher and shannon. *Information and Control*, 2(2):101–112, 1959. ISSN 0019-9958. doi: https://doi.org/10.1016/S0019-9958(59)90348-1. URL https://www.sciencedirect.com/science/article/pii/S0019995859903481.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: A circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=NpsVSN6o4ul.

Yihan Wang, Jatin Chauhan, Wei Wang, and Cho-Jui Hsieh. Universality and limitations of prompt tuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 75623–75643. Curran Associates, Inc., 2023b. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/eef6aecfe050b556c6a48d9c16b15558-Paper-Conference.pdf.

Colin Wei, Yining Chen, and Tengyu Ma. Statistically meaningful approximation: a case study on approximating turing machines with transformers. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 12071–12083. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/4ebf1d74f53ece08512a23309d58df89-Paper-Conference.pdf.

Jeffrey Willette, Heejun Lee, and Sung Ju Hwang. Delta attention: Fast and accurate sparse attention inference by delta correction. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=LZrRvYBqsJ.

Thomas Wolf, Lysandre Debut, Victor Sanh, et al. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6/.

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. Duoattention: Efficient long-context llm inference with retrieval and streaming heads, 2024a. URL https://arxiv.org/abs/2410.10819.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks, 2024b. URL https://arxiv.org/abs/2309.17453.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Chenxiao Yang, Nathan Srebro, and Zhiyuan Li. Tight sample complexity of transformers, 2026. URL https://arxiv.org/abs/2606.09731.

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

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Hf17y6u9BC.

A Details of the Information Propagation Examples

This appendix gives the experimental details for Section 3. Appendix A.1 describes the models and attention masks, Appendix A.2 specifies the activation intervention, Appendices A.3, A.4 report the two examples, and Appendix A.5 discusses limitations and controls.

A.1 Experimental Setup

The traffic-light example uses the pretrained Qwen/Qwen2.5-0.5B (Qwen Team, 2024) checkpoint, which has 24 decoder layers. The additional example below uses Qwen/Qwen3-8B (Yang et al., 2025) which has 36 decoder layers. Both checkpoints are frozen and we compare each checkpoint under full attention and under a causal SWA mask imposed in every layer at inference time. The models were both pretrained with full attention and were not fine-tuned after imposing the local mask. All runs use eager attention and disable the key–value cache. Qwen2.5-0.5B is evaluated with float32 hidden states and Qwen3-8B with float16 hidden states, while its final logits and all reported margins are converted to float32.

We use the paper's window convention throughout: W is the number of preceding positions visible to a token, as defined in Section 2. The implementation's configured span also counts the current position, so it is set to $W + 1$. If $\mathbf{t}_{\mathrm{prev}}$ and $\mathbf{t}_{\mathrm{last}}$ are separated by Δ positions, we report

$$k = \left\lceil \frac{\Delta}{W} \right\rceil \tag{2}$$

as the minimum number of SWA attention steps between them. When $\mathbf{t}_{\mathrm{prev}}$ lies outside the window of $\mathbf{t}_{\mathrm{last}}$, k steps require at least $k - 1$ intermediate relay positions.

A.2 Activation-patching Implementation

Section 3 defines the hidden state $\mathbf{h}^i_{r,p}$, the directional output margins m^i, and the recovery score $R_{r,p}$. Here we specify the intervention used to compute that score. We first run both prompts without intervention and record their hidden states. For each layer $r \in \{0, \ldots, \ell - 1\}$ and position p, we rerun prompt 1 again and replace only the stored vector entering the layer r at position p with its prompt-2 counterpart:

$$\widetilde{\mathbf{h}}^{(p)}_{r,q} = \begin{cases} \mathbf{h}^2_{r,p}, & q = p, \\ \mathbf{h}^1_{r,q}, & q \neq p. \end{cases}$$

The replacement is made before the layer's input RMS normalization, attention, residual updates, and MLP. All later computations are left unchanged and the resulting margin of the next token $m_{r,p}$ is converted to $R_{r,p}$, recall the definition from Section 3,

$$R_{r,p} = \frac{m^1 - m_{r,p}}{m^1 - m^2}.$$

We repeat the intervention for every (r, p) to produce the heatmap. In the heatmaps, $R_{r,p}$ is clipped to $[-1, 1]$ and displayed using $\mathrm{sign}(R_{r,p})|R_{r,p}|^2$ to de-emphasize moderate effects.

A.3 Traffic-Light Example

The matched prompts are

```
The traffic light was green/red.
The driver approached the intersection and decided to
```

They contain 14 tokens and the positions of $\mathbf{t}_{\mathrm{prev}}$ and $\mathbf{t}_{\mathrm{last}}$ are 4 and 13, respectively, so $\Delta = 9$. To match the intervention direction in Figure 1, we index the `red` prompt and `stop` continuation by $i = 1$, and the `green` prompt and `go` continuation by $i = 2$. We therefore insert the `green`-run state $\mathbf{h}^2_{r,p}$ into the otherwise unchanged `red` run.

Table 2 reports the directional margins m^1 and m^2 from Section 3, their absolute separation, and the unrestricted vocabulary ranks of the intended continuations. Under both SWA masks, the two

Attention	W	k	m^1	m^2	$\lvert m^1 - m^2 \rvert$	ranks $1/2$
Full attention	–	–	1.470	0.207	1.263	$1/3$
SWA	5	2	0.172	-1.108	1.280	$8/15$
SWA	3	3	0.269	-1.628	1.897	$20/7$

Table 2: Output margins for the traffic-light example. "ranks $1/2$" denotes the rank of `stop` under the red prompt followed by the rank of `go` under the green prompt. Ranks are computed over the complete vocabulary.

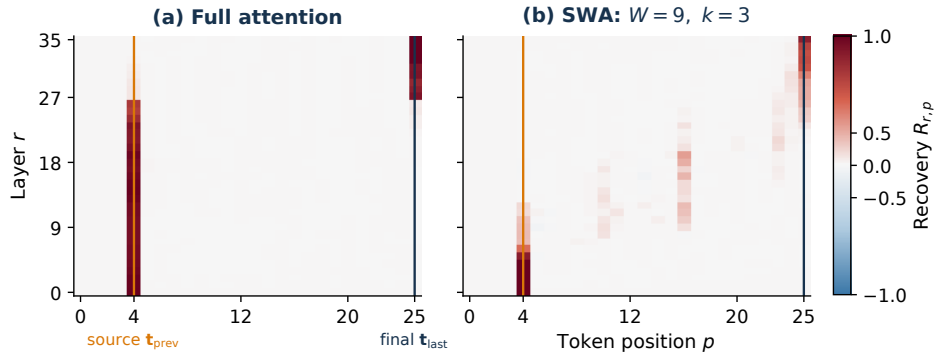


Figure 5: **A behaviorally successful longer relay.** The positions of $\mathbf{t}_{\text{prev}}$ and $\mathbf{t}_{\text{last}}$ are separated by $\Delta = 21$, with $W = 9$ and $k = 3$. Red denotes movement towards the `closed`-run endpoint and blue movement in the opposite direction. We use the same clipping and color mapping as in the main text heatmaps.

prompts have the intended pairwise ordering but under full attention, both prompts prefer `stop`, although changing $\mathbf{t}_{\text{prev}}$ still moves the margin in the expected direction.

The unrestricted top prediction is `stop` for both full-attention prompts, `approach` for both $W = 5$ prompts, and `decided` for both $W = 3$ prompts. The example is therefore used to illustrate intermediate causal recoverability, not as a successful unrestricted completion. A successful completion is shown in the following Appendix subsection.

A.4 Longer Gate Example

The prompt pair we used for a longer and successful relay is the following

```
The garden gate was open/closed.
It did not move while Elena briefly checked the mailbox.
When she looked again, the gate was still
```

The prompts contain 26 tokens and the positions of $\mathbf{t}_{\text{prev}}$ and $\mathbf{t}_{\text{last}}$ are 4 and 25, respectively, so $\Delta = 21$. We set $W = 9$, so $k = 3$ and at least two intermediate token positions are required. Here $i = 1$ denotes the `open` prompt and continutation, while $i = 2$ denotes the `closed` prompt continuation. We therefore insert the `closed`-run state into the `open` run.

Under SWA, the unmodified margins are $m^1 = 1.723$ and $m^2 = -2.891$, giving $|m^1 - m^2| = 4.614$ logits. Under full attention, they are $m^1 = 7.492$ and $m^2 = -7.212$, giving $|m^1 - m^2| = 14.704$ logits. In both attention conditions the intended continuation is ranked first over the complete vocabulary for both prompts.

A.5 Interpretation and Limitations

A large $R_{r,p}$ means that replacing one hidden state is sufficient to restore part of the prompt-2 output effect. High-recovery intermediate states are therefore candidate relays of the influence of $\mathbf{t}_{\text{prev}}$. Across the two examples, recovery under full attention is concentrated mainly near $\mathbf{t}_{\text{prev}}$ and $\mathbf{t}_{\text{last}}$. When SWA places $\mathbf{t}_{\text{prev}}$ outside the local window of $\mathbf{t}_{\text{last}}$, substantial recovery also appears at

intermediate positions. The maps thus give a causal, layer-by-position view of how a distant effect of an earlier token can remain available to the prediction of $\mathbf{t}_{\text{next}}$ across successive local windows.

The examples do not identify a unique route or directly measure how much information a hidden state contains. Hidden states are contextual, so punctuation and function words can also carry information from earlier parts of the sentence. The behavior was not consistent across all prompts and window sizes: some produced less structured maps, and under forced SWA the top prediction did not always match the intended continuation. These selected examples therefore mainly serve an illustration purpose.

B Proof of Theorem 5

We first introduce the required notations in Section B.1. We then provide the sketch of the construction for a single attention head in Section B.2. This construction is based on the fundamental block defined in Section B.3, which then allows to implement both the shift block (Section B.4.1) and the computation block (Section B.4.2) as shown in Section B.4.

General attention can then be approximated by summing the approximations for every head.

B.1 Notations

We add at position 0 a sentinel state always fixed to $\overline{\mathbf{m}_0} = \mathbf{0}$ and work on positions $1, \dots, n$.

The SWA hidden state at position $t > 0$ is

$$\overline{\mathbf{m}_t} = [\mathbf{m}_t, \mathbf{s}_t, t] \in \mathbb{R}^{2Nd+1},$$

where $\mathbf{m}_t \in \mathbb{R}^{Nd}$ is the main state and $\mathbf{s}_t \in \mathbb{R}^{Nd}$ is the scratch space.

The main state is decomposed as

$$\mathbf{m}_t = [\mathbf{m}_t^0, \mathbf{m}_t^1, \dots, \mathbf{m}_t^{N-1}], \qquad \text{where } \mathbf{m}_t^j \in \mathbb{R}^d.$$

At the boundary of every SWA block, we maintain the invariant

$$\mathbf{s}_t = \mathbf{0}.$$

B.2 Sketch of construction

The approximation of a regular single-head transformer layer using SWA is as follows:

1. Initialize
$$\mathbf{m}_t = [\mathbf{h}_t, \mathbf{0}, \dots, \mathbf{0}], \qquad \mathbf{s}_t = \mathbf{0}.$$
2. Apply the shift block from Section B.4.1 $N-1$ times to obtain for all $t \le n$
$$[\mathbf{m}_t^0, \mathbf{m}_t^1, \dots, \mathbf{m}_t^{t-1}] = [\mathbf{h}_t, \mathbf{h}_{t-1}, \dots, \mathbf{h}_1].$$
3. Apply the computation block from Section B.4.2 to obtain
$$\mathbf{m}_t^+ \simeq [\mathbf{h}_t^+, \mathbf{0}, \dots, \mathbf{0}].$$

The number of SWA blocks used is

$$(N-1) + 1 = N.$$

B.3 Fundamental Building Block

Let $G = (G_t)_{t \in [n]}$ be any list of continuous maps satisfying

$$G_t : K \subset \mathbb{R}^{Nd} \times \mathbb{R}^{Nd} \to \mathbb{R}^{Nd}$$

with K compact, $G_0(\mathbf{0}, \mathbf{0}) = \mathbf{0}$. There exists a SWA block $\mathcal{B}_G$ implementing $\mathbf{m}_t \mapsto G_t(\mathbf{m}_{t-1}, \mathbf{m}_t)$ while keeping $\mathbf{m}_0 = \mathbf{0}$ on the main state. Formally, for all $\varepsilon > 0$,

$$\begin{aligned}
&\mathcal{B}_G(\mathbf{0}, \overline{\mathbf{m}_1}, \dots, \overline{\mathbf{m}_n}) = (\mathbf{0}, \overline{\mathbf{m}_1^+}, \dots, \overline{\mathbf{m}_n^+}),\\
&\text{with} \qquad \overline{\mathbf{m}_t^+} = [\mathbf{m}_t^+, \mathbf{0}, t],\\
&\text{and} \qquad \|\mathbf{m}_t^+ - G_t(\mathbf{m}_{t-1}, \mathbf{m}_t)\|_\infty \le \varepsilon \qquad \text{for all } t \in [n].
\end{aligned}$$

B.3.1 SWA IMPLEMENTATION

Attention Component Given

$$\overline{\mathbf{m}_t} = [\mathbf{m}_t, \mathbf{0}, t],$$

we want a single SWA block to expose both $\mathbf{m}_{t-1}$ and $\mathbf{m}_t$ to the pointwise map.

Choose the SWA query and key maps to be zero:

$$\mathbf{Q}_{\text{swa}} = \mathbf{0}, \qquad \mathbf{K}_{\text{swa}} = \mathbf{0}.$$

Then the two softmax weights on $\{t-1, t\}$ are exactly

$$\alpha^{\text{swa}}_{t,t-1} = \alpha^{\text{swa}}_{t,t} = \frac{1}{2}.$$

Choose the value matrix so that it writes the main state into the scratch coordinates:

$$\mathbf{V}_{\text{swa}}[\mathbf{m}, \mathbf{s}, t] = [\mathbf{0}, \mathbf{m}, 0].$$

Since $\mathbf{s}_{t-1} = \mathbf{s}_t = \mathbf{0}$ at block input, the attention update gives

$$\overline{\mathbf{m}_t^{\sim}} = \overline{\mathbf{m}_t} + \frac{1}{2}[\mathbf{0}, \mathbf{m}_{t-1}, 0] + \frac{1}{2}[\mathbf{0}, \mathbf{m}_t, 0].$$

Therefore after the attention sublayer,

$$\overline{\mathbf{m}_t^{\sim}} = \left[\mathbf{m}_t, \frac{\mathbf{m}_{t-1} + \mathbf{m}_t}{2}, t\right].$$

MLP Component Consider the compact set of states produced by the attention component,

$$\widetilde{K} = \left\{ \left[\mathbf{a}, \frac{\mathbf{a}+\mathbf{b}}{2}, t\right] : (\mathbf{a}, \mathbf{b}) \in K,\ t \in [n] \right\} \cup \{\mathbf{0}\}.$$

On this set, define the ideal main-state residual map

$$F\left(\left[\mathbf{a}, \frac{\mathbf{a}+\mathbf{b}}{2}, t\right]\right) = G_t(\mathbf{a}, \mathbf{b}) - \mathbf{a}, \qquad F(\mathbf{0}) = \mathbf{0}.$$

This map is continuous by continuity of F. By the universal approximation theorem (Lu et al., 2021), for every $\varepsilon > 0$ there is an MLP $\widehat{F}$ such that

$$\sup_{\mathbf{z} \in \widetilde{K}} \|\widehat{F}(\mathbf{z}) - F(\mathbf{z})\|_\infty \le \frac{\varepsilon}{2}.$$

Replacing $\widehat{F}(\mathbf{z})$ by $\widehat{F}(\mathbf{z}) - \widehat{F}(\mathbf{0})$, which only changes its final bias, ensures $\widehat{F}(\mathbf{0}) = \mathbf{0}$ and increases the approximation error to at most ε.

Use this approximation in the pointwise residual map

$$\text{MLP}_G[\mathbf{m}^{\sim}, \mathbf{s}^{\sim}, t] = [\widehat{F}(\mathbf{m}^{\sim}, \mathbf{s}^{\sim}, t), -\mathbf{s}^{\sim}, 0].$$

The affine scratch-space term can be represented exactly (in particular, by a ReLU MLP), while the main-state term is approximated as above. For every $t \in [n]$, the block therefore outputs

$$\overline{\mathbf{m}_t^{+}} = [\mathbf{m}_t^{+}, \mathbf{0}, t], \qquad \text{with } \left\|\mathbf{m}_t^{+} - G_t(\mathbf{m}_{t-1}, \mathbf{m}_t)\right\|_\infty \le \varepsilon,$$

as claimed. The dependence of the required MLP size on ε is discussed in Remark 7.

B.4 FULL CONSTRUCTION

We are now equipped to construct the full Sliding Window approximation, which consists of $N-1$ shift blocks (Section B.4.1) followed by one computation block (Section B.4.2). Each of these blocks is of the form described in Section B.3, for specific functions $G_t : K \to \mathbb{R}^{Nd}$ which we describe below.

B.4.1 SHIFT BLOCK

Define

$$\text{Shift} : \mathbb{R}^{Nd} \times \mathbb{R}^{Nd} \to \mathbb{R}^{Nd}$$

as follows. If

$$\mathbf{m}_t = [\mathbf{m}_t^0, \mathbf{m}_t^1, \ldots, \mathbf{m}_t^{N-1}]$$

and

$$\mathbf{m}_{t-1} = [\mathbf{m}_{t-1}^0, \mathbf{m}_{t-1}^1, \ldots, \mathbf{m}_{t-1}^{N-1}],$$

then

$$\text{Shift}(\mathbf{m}_{t-1}, \mathbf{m}_t) = [\mathbf{m}_t^0, \mathbf{m}_{t-1}^0, \mathbf{m}_{t-1}^1, \ldots, \mathbf{m}_{t-1}^{N-2}].$$

Thus slot 0 stays fixed, and the remaining slots receive data from the previous position.

SWA implementation Apply Block B.3 with

$$G = \text{Shift}\,.$$

Hence one SWA block implements

$$\overline{\mathbf{m}_t^+} = [\text{Shift}(\mathbf{m}_{t-1}, \mathbf{m}_t), \mathbf{0}, t].$$

Remark 12. Although we invoke the generic map G for simplicity, the shift map is affine and can be represented exactly by a SWA block with ReLU-MLP of length 2 and width $4Nd$, as shown in Appendix E.1. We therefore use equality here rather than an approximate equality.

Combining Blocks We claim that after $r \le n-1$ shifts,

$$\mathbf{m}_t^{j,(r)} = \begin{cases} \mathbf{h}_{t-j}, & 0 \le j \le r \text{ and } t-j \ge 1, \\ \mathbf{0}, & \text{otherwise.} \end{cases}$$

Proof. For $r = 0$, this holds as

$$\mathbf{m}_t^{(0)} = [\mathbf{h}_t, \mathbf{0}, \ldots, \mathbf{0}], \qquad t \in [n],$$

and

$$\mathbf{m}_0^{(0)} = \mathbf{0}.$$

Suppose it holds after r shifts. The shift block gives

$$\mathbf{m}_t^{0,(r+1)} = \mathbf{m}_t^{0,(r)}$$

and

$$\mathbf{m}_t^{j+1,(r+1)} = \mathbf{m}_{t-1}^{j,(r)}.$$

Therefore, whenever $0 \le j \le r$ and $t-(j+1) \ge 1$,

$$\mathbf{m}_t^{j+1,(r+1)} = \mathbf{m}_{t-1}^{j,(r)} = \mathbf{h}_{t-1-j} = \mathbf{h}_{t-(j+1)}.$$

Also

$$\mathbf{m}_t^{0,(r+1)} = \mathbf{h}_t.$$

Thus the claim follows by induction. □

Hence after $n-1$ shifts

$$\text{for all } t \le n, \qquad [\mathbf{m}_t^0, \mathbf{m}_t^1, \ldots, \mathbf{m}_t^{t-1}] = [\mathbf{h}_t, \mathbf{h}_{t-1}, \ldots, \mathbf{h}_1]. \tag{3}$$

Note that after $n-1$ shifts, the shift operation doesn't modify the parameters decribed above. Therefore, Equation 3 still holds after $N-1 \ge n-1$ shifts.

B.4.2 COMPUTATION BLOCK

Once the gather phase is complete, position t stores the whole prefix in reverse order. Write

$$\Phi_t(\mathbf{h}_1, \dots, \mathbf{h}_t) = \mathrm{L}(\mathbf{h}_1, \dots, \mathbf{h}_t)_{:,-1} = \mathbf{h}_t^+,$$

the attention map, and define $G_t(\mathbf{m}_{t-1}, \mathbf{m}_t) = g_t(\mathbf{m}_t)$ with

$$g_t([\mathbf{m}^0, \mathbf{m}^1, \dots, \mathbf{m}^N]) = \left[\Phi_t(\mathbf{m}^{t-1}, \mathbf{m}^{t-2}, \dots, \mathbf{m}^0), \mathbf{0}, \dots, \mathbf{0}\right] \qquad \text{for } t \in [n],$$

with $G_0(\mathbf{0}) = \mathbf{0}$.

Applying the Block B.3 with G_t allows to conclude.

C IMPROVING THE HIDDEN DIMENSION

The construction of Appendix B is suboptimal: instead of storing the full hidden states $(\mathbf{h}_1, \dots, \mathbf{h}_t)$ in every $\mathbf{m}_t$, it would be enough to store half of them (for a sliding window of size 1). The sliding window then allows to recover the full information for the final computation block.

We first provide the optimized construction for sliding window of size 1 in Appendix C.1, and then provide the general window length-dependant construction in Appendix C.2.

C.1 COMPRESSION FOR WINDOW SIZE 1

We can reduce the hidden dimension of our construction:

$$\mathbf{m}_t, \mathbf{s}_t \in \mathbb{R}^{\lceil \frac{N}{2} \rceil d}.$$

Thus the hidden dimension becomes

$$2 \left\lceil \frac{N}{2} \right\rceil d + 1.$$

The goal is to maintain, after r parity-shift steps,

$$\mathbf{m}_t^{j,(r)} = \begin{cases} \mathbf{h}_{t-2j}, & 0 \le j \le r \text{ and } t - 2j \ge 1, \\ \mathbf{0}, & \text{otherwise.} \end{cases}$$

Thus position t stores

$$[\mathbf{h}_t, \mathbf{h}_{t-2}, \mathbf{h}_{t-4}, \dots],$$

while position $t-1$ stores

$$[\mathbf{h}_{t-1}, \mathbf{h}_{t-3}, \mathbf{h}_{t-5}, \dots].$$

C.1.1 TWO-BLOCK PARITY SHIFT

The parity shift is implemented by two applications of the fundamental block from Section B.3.

Define two local maps

$$A, B : \mathbb{R}^{\lceil \frac{N}{2} \rceil d} \times \mathbb{R}^{\lceil \frac{N}{2} \rceil d} \to \mathbb{R}^{\lceil \frac{N}{2} \rceil d}$$

as follows. If

$$\mathbf{a} = [\mathbf{a}^0, \dots, \mathbf{a}^{\lceil \frac{N}{2} \rceil - 1}], \qquad \mathbf{b} = [\mathbf{b}^0, \dots, \mathbf{b}^{\lceil \frac{N}{2} \rceil - 1}],$$

then

$$A(\mathbf{a}, \mathbf{b}) = [\mathbf{b}^0, \mathbf{a}^0, \mathbf{a}^1, \dots, \mathbf{a}^{\lceil \frac{N}{2} \rceil - 2}],$$

and

$$B(\mathbf{a}, \mathbf{b}) = [\mathbf{b}^0, \mathbf{a}^1, \mathbf{a}^2, \dots, \mathbf{a}^{\lceil \frac{N}{2} \rceil - 1}].$$

Starting from states $\mathbf{m}_t$, first apply Block B.3 with

$$G = A.$$

This gives intermediate states

$$\mathbf{m}'_t = [\mathbf{m}^0_t, \mathbf{m}^0_{t-1}, \mathbf{m}^1_{t-1}, \ldots, \mathbf{m}^{\lceil \frac{N}{2} \rceil - 2}_{t-1}].$$

Then apply Block B.3 again with

$$G = B.$$

Since

$$\mathbf{m}'_{t-1} = [\mathbf{m}^0_{t-1}, \mathbf{m}^0_{t-2}, \mathbf{m}^1_{t-2}, \ldots, \mathbf{m}^{\lceil \frac{N}{2} \rceil - 2}_{t-2}],$$

we obtain

$$\mathbf{m}^+_t = [\mathbf{m}^0_t, \mathbf{m}^0_{t-2}, \mathbf{m}^1_{t-2}, \ldots, \mathbf{m}^{\lceil \frac{N}{2} \rceil - 2}_{t-2}].$$

Hence one parity-shift step moves information two positions to the right while preserving only the parity class of t.

Remark 13. Although we invoke the generic building block for simplicity, A and B are affine and can be represented exactly by SWA blocks with ReLU-MLP of length 2 and width $4\lceil N/2 \rceil d$, as shown in Appendix E.2. We therefore use equality here rather than an approximate equality.

C.1.2 Combining Parity Shifts

Initialize

$$\mathbf{m}^{(0)}_t = [\mathbf{h}_t, \mathbf{0}, \ldots, \mathbf{0}], \qquad \mathbf{s}^{(0)}_t = \mathbf{0}.$$

After one parity-shift step,

$$\mathbf{m}^{(1)}_t = [\mathbf{h}_t, \mathbf{h}_{t-2}, \mathbf{0}, \ldots, \mathbf{0}].$$

More generally, after r parity shifts,

$$\mathbf{m}^{j,(r)}_t = \begin{cases} \mathbf{h}_{t-2j}, & 0 \le j \le r \text{ and } t - 2j \ge 1, \\ \mathbf{0}, & \text{otherwise.} \end{cases}$$

Indeed, the claim is true for $r = 0$. If it holds after r shifts, then the two-block parity shift gives

$$\mathbf{m}^{0,(r+1)}_t = \mathbf{m}^{0,(r)}_t = \mathbf{h}_t,$$

and, for $j \ge 0$,

$$\mathbf{m}^{j+1,(r+1)}_t = \mathbf{m}^{j,(r)}_{t-2}.$$

Therefore, whenever $0 \le j \le r$ and $t - 2(j+1) \ge 1$,

$$\mathbf{m}^{j+1,(r+1)}_t = \mathbf{m}^{j,(r)}_{t-2} = \mathbf{h}_{t-2-2j} = \mathbf{h}_{t-2(j+1)}.$$

This proves the claim by induction.

After $\lceil \frac{N}{2} \rceil - 1$ parity shifts, position t contains all previous tokens with the same parity:

$$\mathbf{m}_t = [\mathbf{h}_t, \mathbf{h}_{t-2}, \mathbf{h}_{t-4}, \ldots].$$

C.1.3 Computation block

At the end of the parity gather, the full prefix of position t is split between positions t and $t-1$. Specifically, for $i \le t$,

$$\mathbf{h}_i = \begin{cases} \mathbf{m}^{(t-i)/2}_t, & i \equiv t \pmod 2, \\ \mathbf{m}^{(t-1-i)/2}_{t-1}, & i \equiv t-1 \pmod 2. \end{cases}$$

Define the local computation map

$$G_t : \mathbb{R}^{\lceil \frac{N}{2} \rceil d} \times \mathbb{R}^{\lceil \frac{N}{2} \rceil d} \to \mathbb{R}^{\lceil \frac{N}{2} \rceil d}$$

by

$$G_t(\mathbf{m}_{t-1}, \mathbf{m}_t) = [\Phi_t(\mathbf{x}_1, \ldots, \mathbf{x}_t), \mathbf{0}, \ldots, \mathbf{0}],$$

where

$$\mathbf{x}_i = \begin{cases} \mathbf{m}^{(t-i)/2}_t, & i \equiv t \pmod 2, \\ \mathbf{m}^{(t-1-i)/2}_{t-1}, & i \equiv t-1 \pmod 2. \end{cases}$$

Then apply the fundamental block from Section B.3 with G_t.

This outputs

$$\mathbf{m}_t^+ = [\mathbf{h}_t^+, \mathbf{0}, \ldots, \mathbf{0}],$$

up to the approximation error of the fundamental block.

Thus one full attention layer can be approximated using hidden dimension $2\left\lceil \frac{N}{2} \right\rceil d + 1$.

C.2 Compression for general window size

Now consider sliding-window attention with window size W, and write $w = W + 1$. We can generalize the parity construction above.

We use a main state $\mathbf{m}_t \in \mathbb{R}^{\lceil \frac{N}{w} \rceil d}$, with decomposition

$$\mathbf{m}_t = [\mathbf{m}_t^0, \ldots, \mathbf{m}_t^{\lceil \frac{N}{w} \rceil - 1}], \qquad \mathbf{m}_t^j \in \mathbb{R}^d.$$

And we decompose the scratch state as

$$\mathbf{s} = [\mathbf{s}^0, \ldots, \mathbf{s}^{w-1}], \qquad \mathbf{s}^r \in \mathbb{R}^{\lceil \frac{N}{w} \rceil d}.$$

The hidden dimension stays of order Nd:

$$(w+1)\left\lceil \frac{N}{w} \right\rceil d + 1,$$

while the number of layers is of order $4\frac{N}{w}$ ($4\lceil \frac{N}{w} \rceil - 2$ to be exact).

C.2.1 Fundamental Building Block for Arbitrary Window Size

Let $G = (G_t)_{t \in [n]}$ be any list of continuous maps satisfying

$$G_t : K \subset (\mathbb{R}^{\lceil \frac{N}{w} \rceil d})^w \to \mathbb{R}^{\lceil \frac{N}{w} \rceil d},$$

where K is compact, and

$$G_0(\mathbf{0}, \ldots, \mathbf{0}) = \mathbf{0}.$$

There exists a two-layer SWA block $\mathcal{B}_G$ implementing

$$\mathbf{m}_t \mapsto G_t(\mathbf{m}_{t-W}, \ldots, \mathbf{m}_{t-1}, \mathbf{m}_t)$$

while keeping the sentinel state fixed to $\mathbf{0}$. We use the convention

$$\mathbf{m}_j = \mathbf{0} \qquad \text{for } j \le 0.$$

Formally, for every $\varepsilon > 0$,

$$\begin{aligned} &\mathcal{B}_G(\mathbf{0}, \overline{\mathbf{m}_1}, \ldots, \overline{\mathbf{m}_n}) = (\mathbf{0}, \overline{\mathbf{m}_1^+}, \ldots, \overline{\mathbf{m}_n^+}), \\ &\text{with} \qquad \overline{\mathbf{m}_t^+} = [\mathbf{m}_t^+, \mathbf{0}, t], \\ &\text{and} \qquad \left\| \mathbf{m}_t^+ - G_t(\mathbf{m}_{t-W}, \ldots, \mathbf{m}_t) \right\|_\infty \le \varepsilon \qquad \text{for all } t \in [n]. \end{aligned}$$

Sketch of Construction

- The first SWA layer produces $\overline{\mathbf{m}_{t-i}} = [\mathbf{0}, \ldots, \mathbf{0}, \underbrace{\mathbf{m}_{t-i}}_{\text{at position } i \text{ modulo } w}, \mathbf{0}, \ldots, \mathbf{0}, t-i]$.
- After the attention component of the second layer, the final state is (up to permutation and constant coefficients) $\overline{\mathbf{m}_t} = [\mathbf{0}, \mathbf{m}_t, \ldots, \mathbf{m}_{t-W}, t]$.
- The final MLP component implements an approximation of G_t.

First SWA Layer Define $\rho(t) := t \bmod w$, and let $\mathbf{E}_r : \mathbb{R}^{\lceil \frac{N}{w} \rceil d} \to \mathbb{R}^{\lceil \frac{N}{w} \rceil dw}$ denote the canonical injection into the r-th scratch block.

Attention Component

Choose the value matrix to be zero:

$$\mathbf{V}^{(1)}_{\text{swa}} = \mathbf{0}.$$

The attention component is therefore identically zero, and the residual attention update leaves the input unchanged:

$$\overline{\mathbf{m}^{\sim}_t} = [\mathbf{m}_t, \mathbf{0}, t].$$

MLP Component

Use the position-dependent affine residual map

$$\text{MLP}^{(1)}[\mathbf{m}, \mathbf{0}, t] = [-\mathbf{m}, \mathbf{E}_{\rho(t)}\mathbf{m}, 0].$$

This map is represented exactly by the construction of Appendix E. The output of the first SWA layer is therefore

$$[\mathbf{m}_t, \mathbf{0}, t] + \text{MLP}^{(1)}[\mathbf{m}_t, \mathbf{0}, t] = [\mathbf{0}, \mathbf{E}_{\rho(t)}\mathbf{m}_t, t].$$

Thus each position stores its main state in the scratch block determined by its residue modulo w.

Second SWA Layer For each t, write

$$I_t := \{\max(0, t - W), \ldots, t\}, \qquad q_t := |I_t| = \min(w, t + 1).$$

Attention Component

Choose

$$\mathbf{Q}^{(2)}_{\text{swa}} = \mathbf{0}, \qquad \mathbf{K}^{(2)}_{\text{swa}} = \mathbf{0}.$$

The attention weights over I_t are therefore uniform:

$$\alpha^{\text{swa}}_{ti} = \frac{1}{q_t}, \qquad i \in I_t.$$

Choose the value matrix to preserve the scratch coordinates:

$$\mathbf{V}^{(2)}_{\text{swa}}[\mathbf{m}, \mathbf{s}, t] = [\mathbf{0}, \mathbf{s}, 0].$$

The residual attention update gives

$$\overline{\mathbf{m}^{\sim}_t} = \left[\mathbf{0}, \mathbf{E}_{\rho(t)}\mathbf{m}_t + \frac{1}{q_t}\sum_{i \in I_t} \mathbf{E}_{\rho(i)}\mathbf{m}_i, t\right].$$

The positions in I_t are consecutive and $|I_t| \leq w$. Hence their residues modulo w are pairwise distinct. Consequently, the scratch state contains each vector

$$\mathbf{m}_{t-W}, \ldots, \mathbf{m}_t$$

in a distinct block, with the convention $\mathbf{m}_i = \mathbf{0}$ for $i \leq 0$.

More precisely, every preceding vector $\mathbf{m}_{t-j}$, $1 \leq j \leq \min(W, t)$, has coefficient $1/q_t$, while $\mathbf{m}_t$ has coefficient

$$1 + \frac{1}{q_t}.$$

Since these coefficients and the positions of the corresponding scratch blocks are determined by t, the pointwise map can recover the ordered tuple

$$(\mathbf{m}_{t-W}, \ldots, \mathbf{m}_t).$$

MLP Component

For every $r \in [w-1]$, define the canonical projection $\mathbf{P}_r : \mathbb{R}^{\lceil \frac{N}{w} \rceil dw} \to \mathbb{R}^{\lceil \frac{N}{w} \rceil d}$ by

$$\mathbf{P}_r\left([\mathbf{s}^0, \ldots, \mathbf{s}^{w-1}]\right) = \mathbf{s}^r, \qquad \mathbf{s}^0, \ldots, \mathbf{s}^{w-1} \in \mathbb{R}^{\lceil \frac{N}{w} \rceil d}.$$

Let $\mathcal{R}_t(\mathbf{s}^\sim) = (\mathbf{m}_{t-W}, \ldots, \mathbf{m}_t)$ denote the map that recovers the correct ordering:

$$\mathcal{R}_t(\mathbf{s}^\sim) = \big(R_{t,W}(\mathbf{s}^\sim), \ldots, R_{t,0}(\mathbf{s}^\sim)\big),$$

where

$$R_{t,j}(\mathbf{s}^\sim) = \begin{cases} \mathbf{0}, & j > t, \\ q_t\, \mathbf{P}_{\rho(t-j)}\mathbf{s}^\sim, & 1 \le j \le \min(W, t), \\ \dfrac{q_t}{q_t+1}\, \mathbf{P}_{\rho(t)}\mathbf{s}^\sim, & j = 0. \end{cases}$$

The map

$$[\mathbf{0}, \mathbf{s}^\sim, t] \mapsto G_t\big(\mathcal{R}_t(\mathbf{s}^\sim)\big)$$

is continuous on the compact set reached by the attention component. By universal approximation, its MLP implementation $\widehat{G}$ can be chosen to have uniform error at most ε and to vanish at the sentinel, using the same recentering argument as in Section B.3. Set

$$\mathrm{MLP}^{(2)}[\mathbf{0}, \mathbf{s}^\sim, t] = [\widehat{G}(\mathbf{s}^\sim, t), -\mathbf{s}^\sim, 0].$$

Thus

$$\overline{\mathbf{m}_t^+} = [\mathbf{m}_t^+, \mathbf{0}, t], \qquad \big\|\mathbf{m}_t^+ - G_t(\mathbf{m}_{t-W}, \ldots, \mathbf{m}_t)\big\|_\infty \le \varepsilon,$$

which proves the result.

C.2.2 Residue Shift

A residue shift moves the information stored at position $t-w$ to position t. Since $t-w$ does not belong to the window $\{t-W, \ldots, t\}$, we use two applications of the fundamental block from Section C.2.1.

First, define

$$A : (\mathbb{R}^{\lceil \frac{N}{w} \rceil d})^w \to \mathbb{R}^{\lceil \frac{N}{w} \rceil d}$$

by

$$A(\mathbf{m}_{t-W}, \ldots, \mathbf{m}_t) = [\mathbf{m}_t^0, \mathbf{m}_{t-1}^0, \mathbf{m}_{t-1}^1, \ldots, \mathbf{m}_{t-1}^{\lceil \frac{N}{w} \rceil - 2}].$$

Applying the fundamental block with $G = A$ produces intermediate states

$$\mathbf{a}_t = [\mathbf{m}_t^0, \mathbf{m}_{t-1}^0, \mathbf{m}_{t-1}^1, \ldots, \mathbf{m}_{t-1}^{\lceil \frac{N}{w} \rceil - 2}].$$

Next, define

$$B : (\mathbb{R}^{\lceil \frac{N}{w} \rceil d})^w \to \mathbb{R}^{\lceil \frac{N}{w} \rceil d}$$

by

$$B(\mathbf{a}_{t-W}, \ldots, \mathbf{a}_t) = [\mathbf{a}_t^0, \mathbf{a}_{t-W}^1, \mathbf{a}_{t-W}^2, \ldots, \mathbf{a}_{t-W}^{\lceil \frac{N}{w} \rceil - 1}].$$

Apply the second fundamental block with $G = B$. Since

$$\mathbf{a}_{t-W} = [\mathbf{m}_{t-W}^0, \mathbf{m}_{t-w}^0, \mathbf{m}_{t-w}^1, \ldots, \mathbf{m}_{t-w}^{\lceil \frac{N}{w} \rceil - 2}],$$

the second fundamental block yields

$$\mathbf{m}_t^+ = [\mathbf{m}_t^0, \mathbf{m}_{t-w}^0, \mathbf{m}_{t-w}^1, \ldots, \mathbf{m}_{t-w}^{\lceil \frac{N}{w} \rceil - 2}].$$

Thus one residue shift preserves the residue class of t modulo w.

Starting from

$$\mathbf{m}_t^{(0)} = [\mathbf{h}_t, \mathbf{0}, \ldots, \mathbf{0}],$$

we obtain after r residue shifts

$$\mathbf{m}_t^{j,(r)} = \begin{cases} \mathbf{h}_{t-wj}, & 0 \le j \le r \text{ and } t - wj \ge 1, \\ \mathbf{0}, & \text{otherwise.} \end{cases}$$

Indeed, one residue shift satisfies

$$\mathbf{m}_t^{0,(r+1)} = \mathbf{m}_t^{0,(r)}$$

and

$$\mathbf{m}_t^{j+1,(r+1)} = \mathbf{m}_{t-w}^{j,(r)}.$$

Consequently, after $\lceil \frac{N}{w} \rceil - 1$ residue shifts, position t contains

$$[\mathbf{h}_t, \mathbf{h}_{t-w}, \mathbf{h}_{t-2w}, \dots].$$

Since each residue shift uses two applications of the fundamental block, and each fundamental block contains two SWA layers, one residue shift requires four SWA layers.

Remark 14. We omit the ε error because this block can be represented exactly with SWA with two-layer ReLU MLPs of width at most $4NwD$, as shown in Appendix E.3. Alternatively, we can use Block C.2.1: since the number of blocks is finite and all maps are uniformly continuous on the relevant compact sets, the MLP approximation errors can be chosen sufficiently small so that the final error is arbitrarily small.

C.2.3 Computation block

At the end of the residue gather, the full prefix of t is distributed across the visible window

$$t - W, \dots, t.$$

For $i \le t$, there is a unique pair (r, j) with

$$r \in \{0, \dots, W\}, \qquad j \ge 0, \qquad i = t - r - wj.$$

Equivalently,

$$r \equiv t - i \pmod{w}, \qquad j = \frac{t - r - i}{w}.$$

The token $\mathbf{h}_i$ is stored at position $t - r$ in slot j:

$$\mathbf{h}_i = \mathbf{m}_{t-r}^{j}.$$

Thus the computation map at position t is the continuous map

$$G_t(\mathbf{m}_{t-W}, \dots, \mathbf{m}_t) = [\Phi_t(\mathbf{x}_1, \dots, \mathbf{x}_t), \mathbf{0}, \dots, \mathbf{0}],$$

where, for each $i \le t$,

$$\mathbf{x}_i = \mathbf{m}_{t-r(i)}^{j(i)}$$

with

$$r(i) \equiv t - i \pmod{w}, \qquad r(i) \in \{0, \dots, W\}, \qquad j(i) = \frac{t - r(i) - i}{w}.$$

Using block C.2.1, this computation map can be approximated uniformly on compact sets.

D Optimality of the Depth Scaling

We show that the $\Theta(N/W)$ depth scaling of Appendix C.2 is optimal. The argument is simply that information can travel at most W positions to the right in each SWA layer.

Lemma 15 (Receptive field of SWA)**.** *In an ℓ-layer causal sliding-window transformer with window size W, the output at position t depends only on the input states at positions*

$$\max(1, t - W\ell), \dots, t.$$

This holds regardless of the hidden dimension, the number of attention heads, and the size of the pointwise MLPs.

Proof. The claim follows by induction on ℓ. At depth zero, the state at position t depends only on the input at position t. In the next layer, the attention update at t reads only the previous-layer states at positions $\max(1, t-W), \ldots, t$, while the MLP acts pointwise. By the induction hypothesis, the union of their receptive fields is contained in $\max(1, t-(\ell+1)W), \ldots, t$. □

Corollary 16. *In particular, this implies that the cramming horizon of SWA is upper bounded by $W\ell$. Indeed for cramming, no matter the prompt, the $W\ell+1^{th}$ generated token only depends on the first $W\ell$ generated tokens. Thus SWA can only generate exactly one sentence of length greater than $W\ell$.*

Theorem 17 (Depth lower bound)**.** *For every $d \geq 1$, $N \geq 2$, $W \geq 1$, and $r > 0$, there is a d-dimensional full attention layer on sequences of length N such that no ℓ-layer SWA transformer of window size W can approximate it uniformly on $B^d(0,r)^N$ with error smaller than $r/(2N)$ unless*

$$\ell \geq \left\lceil \frac{N-1}{W} \right\rceil.$$

Consequently, simulating an arbitrary full attention layer requires $\Omega(N/W)$ SWA layers (in the nontrivial regime $W < N$).

Proof. Let $\mathbf{e}_1 \in \mathbb{R}^d$ be the first standard basis vector. Consider the full attention layer with zero queries and keys, identity value map, and zero MLP. Its attention weights are uniform, so at position N its output is

$$\mathbf{h}_N^+ = \mathbf{h}_N + \frac{1}{N}\sum_{i=1}^{N} \mathbf{h}_i.$$

Compare the all-zero input with the input for which $\mathbf{h}_1 = r\mathbf{e}_1$ and $\mathbf{h}_2 = \cdots = \mathbf{h}_N = \mathbf{0}$. Both inputs belong to $B^d(0,r)^N$, and the two target outputs at position N differ by $(r/N)\mathbf{e}_1$, whose ℓ_∞ norm is r/N.

If $W\ell < N-1$, Lemma 15 shows that position N cannot depend on position 1. The SWA transformer therefore produces the same output at position N on both inputs. By the triangle inequality, its error on at least one of them is at least half the distance between the two target outputs, namely $r/(2N)$. Thus uniform approximation below this error requires $W\ell \geq N-1$, which gives the stated bound. □

E Exact Implementation of the Shift Blocks with Two-layer ReLU MLPs

The generic building blocks in Appendices B and C use universal approximation to implement arbitrary continuous maps. Such an argument is needed for the final computation blocks, but not for the shift blocks: their updates are linear in the stored states at each position. We show below that the ordinary shift, the two-block parity shift, and the residue shift can therefore all be implemented exactly, rather than only up to an error ε. Throughout this section, the pointwise component is a two-layer ReLU MLP.

E.1 Exact SWA Implementation of the Shift Block of Section B.4.1

We give explicit weights for the block in Section B.4.1. Define the linear maps $P, R : \mathbb{R}^{Nd} \to \mathbb{R}^{Nd}$ by

$$P[\mathbf{x}^0, \ldots, \mathbf{x}^{N-1}] = [\mathbf{x}^0, \mathbf{0}, \ldots, \mathbf{0}], \qquad R[\mathbf{x}^0, \ldots, \mathbf{x}^{N-1}] = [\mathbf{0}, \mathbf{x}^0, \ldots, \mathbf{x}^{N-2}].$$

Thus the desired update can be written as

$$\mathrm{Shift}(\mathbf{m}_{t-1}, \mathbf{m}_t) = P\mathbf{m}_t + R\mathbf{m}_{t-1}.$$

Attention sublayer. Set the query and key matrices to zero and choose the value matrix, in (main, scratch, position) block form, as

$$\mathbf{Q}_{\mathrm{swa}} = \mathbf{K}_{\mathrm{swa}} = \mathbf{0}, \qquad \mathbf{V}_{\mathrm{swa}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ I_{Nd} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 0 \end{bmatrix}.$$

The attention weights on positions $t-1$ and t are both exactly $1/2$. Consequently, after the attention residual connection,

$$\widetilde{\mathbf{m}}_t = \left[\mathbf{m}_t, \frac{\mathbf{m}_{t-1} + \mathbf{m}_t}{2}, t\right].$$

MLP sublayer. Use the following linear residual map on the state produced by attention:

$$F[\mathbf{m}, \mathbf{s}, t] = [(P - I_{Nd} - R)\mathbf{m} + 2R\mathbf{s}, -\mathbf{s}, 0].$$

This map is implemented exactly by a two-layer ReLU MLP. To see this, let

$$A = \begin{bmatrix} P - I_{Nd} - R & 2R \\ \mathbf{0} & -I_{Nd} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{(2Nd+1)\times 2Nd}, \qquad E = [I_{2Nd} \quad \mathbf{0}],$$

and choose zero biases together with

$$W_1 = \begin{bmatrix} E \\ -E \end{bmatrix} \in \mathbb{R}^{4Nd\times(2Nd+1)}, \qquad W_2 = [A \quad -A] \in \mathbb{R}^{(2Nd+1)\times 4Nd}.$$

Since $\mathrm{ReLU}(x) - \mathrm{ReLU}(-x) = x$ coordinatewise,

$$W_2\,\mathrm{ReLU}(W_1[\mathbf{m}, \mathbf{s}, t]) = A[\mathbf{m}, \mathbf{s}] = F[\mathbf{m}, \mathbf{s}, t].$$

The MLP therefore has length 2 and width $4Nd$.

Finally, adding the MLP residual gives

$$\begin{aligned} \widetilde{\mathbf{m}}_t + F(\widetilde{\mathbf{m}}_t) &= [P\mathbf{m}_t + R(2\mathbf{s}_t - \mathbf{m}_t), \mathbf{0}, t] \\ &= [\mathrm{Shift}(\mathbf{m}_{t-1}, \mathbf{m}_t), \mathbf{0}, t]. \end{aligned}$$

All maps and biases vanish at the origin, so the sentinel remains fixed. Hence this single SWA layer implements Block B.4.1 exactly at every position, including the left boundary.

E.2 Exact SWA Implementation of the Two-block Parity Shift of Section C.1.1

We now give explicit weights for the two local maps in Section C.1.1. Write

$$q = \left\lceil \frac{N}{2} \right\rceil, \qquad D = qd,$$

so that the main and scratch states belong to $\mathbb{R}^D$. Let $P, R : \mathbb{R}^D \to \mathbb{R}^D$ be the slot maps from the preceding subsection, now with q slots, and let

$$Q = I_D - P.$$

Thus Q deletes slot zero and preserves all other slots. The two maps of Section C.1.1 are

$$A(\mathbf{a}, \mathbf{b}) = P\mathbf{b} + R\mathbf{a}, \qquad B(\mathbf{a}, \mathbf{b}) = P\mathbf{b} + Q\mathbf{a}.$$

Both are instances of the linear local map

$$G_L(\mathbf{a}, \mathbf{b}) = P\mathbf{b} + L\mathbf{a},$$

with $L = R$ for the first block and $L = Q$ for the second. We first give an exact one-layer SWA implementation of G_L and then compose the two choices of L.

One-layer implementation of G_L. At the input to either block, write the state as $[\mathbf{b}, \mathbf{0}, t]$, with the preceding position containing $[\mathbf{a}, \mathbf{0}, t-1]$. Use

$$\mathbf{Q}^{(L)}_{\mathrm{swa}} = \mathbf{K}^{(L)}_{\mathrm{swa}} = \mathbf{0}, \qquad \mathbf{V}^{(L)}_{\mathrm{swa}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ I_D & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 0 \end{bmatrix}.$$

After the attention residual connection, the state is

$$[\mathbf{b}, \mathbf{s}, t], \qquad \mathbf{s} = \frac{\mathbf{a} + \mathbf{b}}{2}.$$

The required MLP residual is therefore

$$F_L[\mathbf{b}, \mathbf{s}, t] = [(P - I_D - L)\mathbf{b} + 2L\mathbf{s}, -\mathbf{s}, 0],$$

because $2\mathbf{s} - \mathbf{b} = \mathbf{a}$.

For completeness, define

$$C_L = \begin{bmatrix} P - I_D - L & 2L \\ \mathbf{0} & -I_D \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{(2D+1)\times 2D}, \qquad E_D = [I_{2D} \quad \mathbf{0}].$$

The two-layer ReLU MLP with zero biases and weights

$$W_{1,L} = \begin{bmatrix} E_D \\ -E_D \end{bmatrix} \in \mathbb{R}^{4D\times(2D+1)}, \qquad W_{2,L} = [C_L \quad -C_L] \in \mathbb{R}^{(2D+1)\times 4D}$$

implements F_L exactly. Indeed,

$$W_{2,L}\,\mathrm{ReLU}\big(W_{1,L}[\mathbf{b}, \mathbf{s}, t]\big) = C_L[\mathbf{b}, \mathbf{s}] = F_L[\mathbf{b}, \mathbf{s}, t].$$

Adding the MLP residual yields

$$[\mathbf{b}, \mathbf{s}, t] + F_L[\mathbf{b}, \mathbf{s}, t] = [P\mathbf{b} + L\mathbf{a}, \mathbf{0}, t] = [G_L(\mathbf{a}, \mathbf{b}), \mathbf{0}, t].$$

Thus each local map is implemented by one SWA layer whose MLP has length 2 and width $4D = 4\lceil N/2\rceil d$.

Composition of the two blocks. For the first SWA layer choose $L = R$. It maps

$$\mathbf{m}_t \longmapsto \mathbf{m}'_t = A(\mathbf{m}_{t-1}, \mathbf{m}_t) = [\mathbf{m}^0_t, \mathbf{m}^0_{t-1}, \ldots, \mathbf{m}^{q-2}_{t-1}].$$

For the second SWA layer choose $L = Q$. Since

$$\mathbf{m}'_{t-1} = [\mathbf{m}^0_{t-1}, \mathbf{m}^0_{t-2}, \ldots, \mathbf{m}^{q-2}_{t-2}],$$

this layer maps

$$\begin{aligned} \mathbf{m}'_t &\longmapsto B(\mathbf{m}'_{t-1}, \mathbf{m}'_t) \\ &= [\mathbf{m}^0_t, \mathbf{m}^0_{t-2}, \ldots, \mathbf{m}^{q-2}_{t-2}]. \end{aligned}$$

Both layers clear their scratch coordinates, and their weights and biases vanish at the origin. Hence the sentinel remains fixed between the layers, including at the left boundary. The two-layer composition therefore implements the two-block parity shift exactly.

E.3 Exact SWA Implementation of the Residue Shift of Section C.2.2

We finally make the four-layer residue-shift construction explicit. Write

$$w = W + 1, \qquad p = \left\lceil \frac{N}{w} \right\rceil, \qquad D = pd.$$

The main state belongs to $\mathbb{R}^D$, the scratch state consists of w blocks in $\mathbb{R}^D$, and the total hidden dimension is $(w+1)D + 1$. Let $P, R : \mathbb{R}^D \to \mathbb{R}^D$ preserve slot zero and shift slots one place to the right, respectively, and set $Q = I_D - P$. Then the two maps in Section C.2.2 are

$$A(\mathbf{m}_{t-W}, \ldots, \mathbf{m}_t) = P\mathbf{m}_t + R\mathbf{m}_{t-1}, \qquad B(\mathbf{a}_{t-W}, \ldots, \mathbf{a}_t) = P\mathbf{a}_t + Q\mathbf{a}_{t-W}.$$

Each map is implemented by the following two-layer window-gathering block.

Exact position gates. The construction is required only on the compact state set used in Section C.2.1. Choose $M > 0$ bounding every coordinate that enters a pointwise map, and choose $C > 2M$. For a vector $\mathbf{x}$ with $\|\mathbf{x}\|_\infty \le M$, define

$$\begin{aligned} H_k(\mathbf{x}, t) &= \frac{1}{2}\Big(\mathrm{ReLU}\big(\mathbf{x} + C(t - k + \tfrac{1}{2})\mathbf{1}\big) - \mathrm{ReLU}\big(-\mathbf{x} + C(t - k + \tfrac{1}{2})\mathbf{1}\big)\Big), \\ \Gamma_k(\mathbf{x}, t) &= H_k(\mathbf{x}, t) - H_{k+1}(\mathbf{x}, t). \end{aligned}$$

At integer positions $t \in \{0, \dots, N\}$,

$$\Gamma_k(\mathbf{x}, t) = \begin{cases} \mathbf{x}, & t = k, \\ \mathbf{0}, & t \neq k. \end{cases}$$

Indeed, $H_k(\mathbf{x}, t)$ equals $\mathbf{0}$ for $t < k$ and $\mathbf{x}$ for $t \geq k$. Each Γ_k uses four ReLU units per input coordinate, so any map of the form

$$(\mathbf{x}, t) \longmapsto \sum_{k=1}^{N} L_k \Gamma_k(\mathbf{x}, t)$$

is represented exactly at the relevant integer positions by a two-layer ReLU MLP of width at most $4N \dim(\mathbf{x})$. If different intermediate states have different bounds, we choose C using their maximum. Notice also that all gates vanish at the sentinel $(\mathbf{0}, 0)$.

First layer: residue routing. For $r \in \{0, \dots, w-1\}$, let $E_r : \mathbb{R}^D \to \mathbb{R}^{wD}$ inject a vector into scratch block r, and write $\rho(k) = k \bmod w$. Set the attention value matrix to zero. The attention residual therefore leaves $[\mathbf{m}, \mathbf{0}, t]$ unchanged. Use the exact position-gated MLP residual

$$F^{\text{route}}[\mathbf{m}, \mathbf{0}, t] = \sum_{k=1}^{N} \left[-\Gamma_k(\mathbf{m}, t), E_{\rho(k)} \Gamma_k(\mathbf{m}, t), 0 \right].$$

At position $t \in [N]$, the output of the first SWA layer is exactly

$$[\mathbf{m}_t, \mathbf{0}, t] + F^{\text{route}}[\mathbf{m}_t, \mathbf{0}, t] = [\mathbf{0}, E_{\rho(t)} \mathbf{m}_t, t].$$

This MLP has length 2 and width at most $4ND$.

Second layer: window recovery. Set the queries and keys to zero, and choose the value matrix

$$\mathbf{V}_{\text{swa}}^{\text{gather}} \begin{bmatrix} \mathbf{m} \\ \mathbf{s} \\ t \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{s} \\ 0 \end{bmatrix}.$$

For

$$I_t = \{\max(0, t - W), \dots, t\}, \qquad q_t = |I_t| = \min(w, t + 1),$$

the zero queries and keys give the exact uniform weight $1/q_t$. Hence the scratch state after the attention residual is

$$\mathbf{u}_t = E_{\rho(t)} \mathbf{m}_t + \frac{1}{q_t} \sum_{i \in I_t} E_{\rho(i)} \mathbf{m}_i.$$

Let $\Pi_r : \mathbb{R}^{wD} \to \mathbb{R}^D$ project onto scratch block r. For $j \in \{0, \dots, W\}$, define the linear recovery map

$$\mathcal{D}_{t,j} \mathbf{u} = \begin{cases} \mathbf{0}, & j > t, \\ q_t \Pi_{\rho(t-j)} \mathbf{u}, & 1 \leq j \leq \min(W, t), \\ \dfrac{q_t}{q_t + 1} \Pi_{\rho(t)} \mathbf{u}, & j = 0. \end{cases}$$

The residues represented in I_t are pairwise distinct, so

$$\mathcal{D}_{t,j} \mathbf{u}_t = \mathbf{m}_{t-j},$$

where missing states are interpreted as zero.

For the first window-gathering block, define

$$L_t^A = P\mathcal{D}_{t,0} + R\mathcal{D}_{t,1},$$

and for the second define

$$L_t^B = P\mathcal{D}_{t,0} + Q\mathcal{D}_{t,W}.$$

For $X \in \{A, B\}$, the second-layer MLP residual is the exact gated map

$$F_X^{\text{recover}}[\mathbf{0}, \mathbf{u}, t] = \sum_{k=1}^{N} \left[L_k^X \Gamma_k(\mathbf{u}, t), -\Gamma_k(\mathbf{u}, t), 0 \right].$$

At position t, only the $k = t$ summand remains. The scratch residual is $-\mathbf{u}_t$, while the main-state residual is $L_t^X \mathbf{u}_t$. Consequently the second SWA layer outputs

$$[L_t^X \mathbf{u}_t, \mathbf{0}, t].$$

The recovery MLP has length 2 and width at most $4NwD$.

Composition of the four SWA layers. Apply the routing and recovery layers first with $X = A$. Since

$$L_t^A \mathbf{u}_t = P\mathbf{m}_t + R\mathbf{m}_{t-1},$$

the resulting main state is

$$\mathbf{a}_t = [\mathbf{m}_t^0, \mathbf{m}_{t-1}^0, \ldots, \mathbf{m}_{t-1}^{p-2}].$$

Apply the same two layers again to the states $\mathbf{a}_t$, now with $X = B$. They output

$$P\mathbf{a}_t + Q\mathbf{a}_{t-W} = [\mathbf{m}_t^0, \mathbf{m}_{t-w}^0, \ldots, \mathbf{m}_{t-w}^{p-2}],$$

because $t - W - 1 = t - w$. The scratch state is zero between all four layers, and the position gates vanish at the sentinel. Thus the sentinel and all left boundary conventions are preserved exactly. The four-layer composition implements the residue shift of Section C.2.2 exactly, using two-layer ReLU MLPs of width at most $4NwD$.

F RoPE Allows to Reduce the hidden dimension

The construction of Appendix C.2 is not optimal. Its main state has only $\lceil N/(W+1) \rceil d$ coordinates, but its fundamental block routes the $W+1$ visible states into different scratch blocks. Consequently, its total hidden dimension remains of order Nd. We show that with rotary position embeddings (RoPE; Su et al., 2023), this routing is unnecessary: one SWA layer can directly copy the state at $t - W$ to position t using $O(\frac{N}{W})$ layers of hidden dimension $O(\frac{Nd}{W})$.

RoPE partitions the query's and key's D-dimensional space into two- dimensional subspaces and rotates each by an angle that depends on the absolute position i.

Definition 18 (Rotary position embedding)**.** Let the query–key dimension be $s = 2m$, and fix angular frequencies $\omega_1, \ldots, \omega_m$. For $\theta \in \mathbb{R}$, define

$$\mathrm{Rot}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \qquad \mathcal{R}_t = \mathrm{diag}\big(\mathrm{Rot}(t\omega_1), \ldots, \mathrm{Rot}(t\omega_m)\big) \in \mathbb{R}^{s\times s}.$$

The tuple $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_m)$ is the RoPE frequency schedule.

Given pre-rotation queries and keys

$$\mathbf{q}_t = \mathbf{Q}\mathbf{h}_t, \qquad \mathbf{k}_i = \mathbf{K}\mathbf{h}_i,$$

RoPE uses

$$\mathbf{q}_t^{\mathrm{R}} = \mathcal{R}_t\mathbf{q}_t, \qquad \mathbf{k}_i^{\mathrm{R}} = \mathcal{R}_i\mathbf{k}_i$$

in the attention logits. Thus, for causal sliding-window attention with

$$I_t = \{\max(0, t - W), \ldots, t\},$$

the attention weights and output are

$$\alpha_{ti}^{\mathrm{R}} = \frac{\exp\big(\mathbf{q}_t^\top \mathcal{R}_{i-t}\mathbf{k}_i\big)}{\sum_{j\in I_t} \exp\big(\mathbf{q}_t^\top \mathcal{R}_{j-t}\mathbf{k}_j\big)}, \qquad i \in I_t, \tag{4}$$

$$\mathrm{Att}_t^{\mathrm{R}} = \mathbf{V}\sum_{i\in I_t} \alpha_{ti}^{\mathrm{R}}\mathbf{h}_i. \tag{5}$$

Only queries and keys are rotated; values are not. The usual optional logit factor $1/\sqrt{s}$ is omitted, consistently with Definition 2, since it can be absorbed into $\mathbf{Q}$ and $\mathbf{K}$ (we write $\mathcal{R}_{i-t} := \mathcal{R}_t^\top \mathcal{R}_i$).

F.1 Selecting one relative offset with RoPE

For each fixed $r \in \{0, \ldots, W\}$ with $r < N$, we construct a RoPE attention head, indexed by a scale $\gamma > 0$, whose weights satisfy

$$\forall t \in [\![r+1, N]\!], \qquad \alpha_{t,t-r}^{(r,\gamma)} \xrightarrow[\gamma\to\infty]{} 1.$$

Hence, on bounded value states, its attention output converges uniformly to the value at position $t-r$. We use this selector with $r=W$ for the shift in Section F.2 and with $r=0,\dots,W-1$ for the computation in Section F.3; when $t-r<1$, a pointwise boundary gate returns zero.

Fix one rotary pair from Definition 18, with frequency ω, and write $R_j := \mathrm{Rot}(j\omega)$. We set all other query and key coordinates to zero. Assume that this pair does not alias any two offsets in a window:

$$\Delta_\omega := \min_{1\le|k|\le W}\big(1-\cos(k\omega)\big) > 0. \tag{6}$$

This mild condition holds, for example, for the usual RoPE pair with $\omega = 1$, since no nonzero integer is an integer multiple of 2π.

Write $\overline{\Delta}_\omega := \min(\Delta_\omega, 1)$. We now construct the selector head explicitly. Let d_{sim} denote the total hidden dimension of the simulating SWA transformer. The scalar c_t is an additional hidden-state coordinate indicating whether position t is a real token:

$$\overline{\mathbf{h}}_t = [\mathbf{h}_t, c_t] \in \mathbb{R}^{d_{\mathrm{sim}}}, \qquad c_t = \begin{cases} 1, & t \in \{1,\dots,n\}, \\ 0, & t = 0, \end{cases} \tag{7}$$

where $t=0$ is the all-zero sentinel and $\mathbf{h}_t$ denotes all the other state coordinates. The input embedding initializes this flag, and every simulating layer preserves it: its value map and pointwise residual update are both zero in the c-coordinate. Thus $c_t = 1$ at every real position in every layer, while the sentinel retains $c_0 = 0$. Let

$$P_c \in \mathbb{R}^{1\times d_{\mathrm{sim}}}, \qquad P_c[\mathbf{h}, c] = c,$$

be the row vector projecting onto this coordinate.

Fix an offset $r \in \{0,\dots,W\}$ and a scale $\gamma > 0$, and define

$$\mathbf{e}_1^{(2)} := [1,0]^\top \in \mathbb{R}^2.$$

Define the pair-local query and key projections by

$$\mathbf{Q}^{(r)}_{\mathrm{pair}} = \gamma \mathbf{e}_1^{(2)} P_c \in \mathbb{R}^{2\times d_{\mathrm{sim}}}, \qquad \mathbf{K}^{(r)}_{\mathrm{pair}} = \gamma R_r \mathbf{e}_1^{(2)} P_c \in \mathbb{R}^{2\times d_{\mathrm{sim}}}. \tag{8}$$

Consequently, the pair-local pre-RoPE query and key at real positions are

$$\mathbf{q}_t^{\mathrm{pair}} = \mathbf{Q}^{(r)}_{\mathrm{pair}} \overline{\mathbf{h}}_t = \gamma \mathbf{e}_1^{(2)} \in \mathbb{R}^2, \qquad \mathbf{k}_i^{\mathrm{pair}} = \mathbf{K}^{(r)}_{\mathrm{pair}} \overline{\mathbf{h}}_i = \gamma R_r \mathbf{e}_1^{(2)} \in \mathbb{R}^2.$$

The purpose of the phase R_r in the key projection is to cancel the relative RoPE phase exactly when $i = t-r$. The attention logit between two real tokens is

$$(R_t \mathbf{q}_t^{\mathrm{pair}})^\top (R_i \mathbf{k}_i^{\mathrm{pair}}) = \gamma^2 (\mathbf{e}_1^{(2)})^\top R_{i-t+r} \mathbf{e}_1^{(2)} = \gamma^2 \cos\big((i-t+r)\omega\big). \tag{9}$$

If $t-r \ge 1$, the unique largest logit in the local window is therefore at $i = t-r$. Every other real token has a logit gap at least $\gamma^2\Delta_\omega$; if the zero sentinel is visible, its logit is zero and its gap is γ^2. If $\alpha_{ti}^{(r,\gamma)}$ denotes the corresponding softmax weight, equation 6 gives the explicit bound

$$\alpha^{(r,\gamma)}_{t,t-r} \ge \frac{1}{1 + W\exp(-\gamma^2\overline{\Delta}_\omega)}. \tag{10}$$

Thus, for vectors $\|\mathbf{h}_i\|_\infty \le r$,

$$\left\| \sum_i \alpha_{ti}^{(r,\gamma)} \mathbf{h}_i - \mathbf{h}_{t-r} \right\|_\infty \le 2rW\exp(-\gamma^2\overline{\Delta}_\omega). \tag{11}$$

The selection is consequently arbitrarily accurate as $\gamma \to \infty$. It is not exact for a finite softmax temperature; all claims below are approximation claims. When $t-r<1$, the desired source does not exist. The pointwise map, which also receives the integer position t, sets the copied vector to zero. Exact gates on the finite position set $\{1,\dots,N\}$ can be implemented as in Appendix E.

F.2 A ONE-LAYER STRIDE-W SHIFT

Set

$$p := \left\lceil \frac{N}{W} \right\rceil, \qquad D := pd, \qquad \mathbf{m}_t = [\mathbf{m}_t^0, \ldots, \mathbf{m}_t^{p-1}] \in \mathbb{R}^D, \qquad \mathbf{m}_t^j \in \mathbb{R}^d \quad (0 \le j < p).$$

Here $t \in \{1, \ldots, n\}$ indexes sequence positions and $j \in \{0, \ldots, p-1\}$ indexes the d-dimensional slots of the main state.

Using the selector of Section F.1 with $r = W$, we construct one RoPE–SWA layer such that, up to an arbitrarily small uniform error,

$$\mathbf{m}_t^{\mathrm{out},j} = \begin{cases} \mathbf{m}_t^{\mathrm{in},0}, & j = 0, \\ \mathbf{m}_{t-W}^{\mathrm{in},j-1}, & 1 \le j < p \text{ and } t > W, \\ \mathbf{0}_d, & 1 \le j < p \text{ and } t \le W. \end{cases}$$

Thus iterating it $p-1$ times from the initialization below yields $\mathbf{m}_t^j = \mathbf{h}_{t-jW}$ whenever $t - jW \ge 1$ and zero otherwise.

We augment the state at position t to

$$\overline{\mathbf{m}}_t = [\mathbf{m}_t, \mathbf{u}_t, \mathbf{Z}_t, A_t, \tau_t, c_t] \in \mathbb{R}^{2D+d+3}, \tag{12}$$

where $\mathbf{u}_t \in \mathbb{R}^D$ is the selector scratch block, $\mathbf{Z}_t \in \mathbb{R}^d$ and $A_t \in \mathbb{R}$ are the later numerator and denominator accumulators, $\tau_t = t$ is the absolute-position coordinate, and c_t is the real-token flag from equation 7. At every layer boundary we maintain $\mathbf{u}_t = \mathbf{0}$. Thus the d_{sim} introduced in the selector construction is, for this simulation, $d_{\mathrm{sim}} = 2D + d + 3$.

For every selector layer, use the explicit value map

$$\mathbf{V}_{\mathrm{sel}}[\mathbf{m}, \mathbf{u}, \mathbf{Z}, A, \tau, c] = [\mathbf{0}_D, \mathbf{m}, \mathbf{0}_d, 0, 0, 0]. \tag{13}$$

In a selector layer with $r = W$, the attention residual maps a layer-boundary state to

$$\widetilde{\overline{\mathbf{m}}}_t = \left[\mathbf{m}_t, \underbrace{\sum_{i \in I_t} \alpha_{ti}^{(W,\gamma)} \mathbf{m}_i}_{\widetilde{\mathbf{u}}_t}, \mathbf{Z}_t, A_t, \tau_t, c_t \right]. \tag{14}$$

For $t > W$, Equation equation 11 gives $\widetilde{\mathbf{u}}_t \approx \mathbf{m}_{t-W}$.

To state the pointwise update on its actual input, define

$$\mathrm{src}_W(\widetilde{\mathbf{u}}, \tau) = \begin{cases} \widetilde{\mathbf{u}}, & \tau > W, \\ \mathbf{0}_D, & \tau \le W, \end{cases} \tag{15}$$

on the integer positions $\tau \in \{0, \ldots, N\}$, and define the desired new main memory

$$S_W(\mathbf{m}, \widetilde{\mathbf{u}}, \tau)^j = \begin{cases} \mathbf{m}^0, & j = 0, \\ \mathrm{src}_W(\widetilde{\mathbf{u}}, \tau)^{j-1}, & 1 \le j \le p-1. \end{cases} \tag{16}$$

Here $S_W(\mathbf{m}, \widetilde{\mathbf{u}}, \tau) \in \mathbb{R}^D$, and the superscript denotes a d-dimensional slot. When $p = 1$, only the $j = 0$ case is present. The MLP is a residual map on the complete post-attention state equation 14:

$$\begin{aligned} &\mathrm{MLP}_{\mathrm{shift}}[\mathbf{m}, \widetilde{\mathbf{u}}, \mathbf{Z}, A, \tau, c] \\ &\quad = [S_W(\mathbf{m}, \widetilde{\mathbf{u}}, \tau) - \mathbf{m}, -\widetilde{\mathbf{u}}, \mathbf{0}_d, 0, 0, 0]. \end{aligned} \tag{17}$$

After the MLP residual, the complete output state is therefore

$$[S_W(\mathbf{m}_t, \widetilde{\mathbf{u}}_t, \tau_t), \mathbf{0}_D, \mathbf{Z}_t, A_t, \tau_t, c_t]. \tag{18}$$

For each real position $t \in \{1, \ldots, n\}$, initialize the complete state as

$$\overline{\mathbf{m}}_t^{(0)} = \big[[\mathbf{h}_t, \mathbf{0}_d, \ldots, \mathbf{0}_d], \mathbf{0}_D, \mathbf{0}_d, 0, t, 1\big], \tag{19}$$

and initialize the sentinel as $\overline{\mathbf{m}}_0^{(0)} = \mathbf{0}$.

Ignoring the arbitrarily small selector errors for the moment, induction on a shows that after a stride shifts,

$$\mathbf{m}_t^{j,(a)} = \begin{cases} \mathbf{h}_{t-jW}, & 0 \le j \le a \text{ and } t - jW \ge 1, \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{20}$$

After $p-1$ layers, position t therefore stores every input whose index is congruent to t modulo W.

F.3 Computing full attention without a large final scratch block

It remains to combine the W residue classes visible from t.

The objective is to construct the softmax denominator

$$A_t^\star = \sum_{i=1}^{t} \exp\big((\mathbf{Q}\mathbf{h}_t)^\top \mathbf{K}\mathbf{h}_i\big)$$

and the analogous value-weighted numerator. After the gather phase, $\mathbf{m}_{t-r}$ contains $\mathbf{h}_{t-r}, \mathbf{h}_{t-r-W}, \ldots$; hence the existing memories among positions $t, t-1, \ldots, t-W+1$ partition the entire prefix of t (nonexistent boundary positions contribute nothing). Starting from $A_t = 0$, we process these memories in order. At round r, attention copies $\mathbf{m}_{t-r}$ into the scratch block at position t, and the pointwise MLP adds to A_t all exponential terms represented in that memory. Thus round 0 adds the terms stored at t, round 1 those stored at $t-1$, and so on, until $A_t = A_t^\star$ after W rounds. The numerator $\mathbf{Z}_t$ is accumulated in parallel by the same procedure.

Consider first one target attention head with matrices $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ from Definition 2. For $r = 0, 1, \ldots, W-1$, use the selector query and key projections equation 8 with this value of r, together with the same value map equation 13. The attention residual therefore writes an approximation of $\mathbf{m}_{t-r}$ into $\mathbf{u}_t$ while preserving all other state blocks. Its pointwise map then preserves $\mathbf{m}_t$, clears $\mathbf{u}_t$, leaves (τ_t, c_t) unchanged, and, for all valid slots, performs

$$A_t \longleftarrow A_t + \sum_{\substack{0 \le j < p \\ t-r-jW \ge 1}} \exp\Big((\mathbf{Q}\mathbf{h}_t)^\top \mathbf{K}\mathbf{m}_{t-r}^j\Big), \tag{21}$$

$$\mathbf{Z}_t \longleftarrow \mathbf{Z}_t + \sum_{\substack{0 \le j < p \\ t-r-jW \ge 1}} \exp\Big((\mathbf{Q}\mathbf{h}_t)^\top \mathbf{K}\mathbf{m}_{t-r}^j\Big)\, \mathbf{V}\mathbf{m}_{t-r}^j. \tag{22}$$

The validity masks depend only on the integer position t. The updates are continuous on the compact state domain and hence can be approximated by the pointwise MLP.

Consequently, after the W selector layers,

$$A_t = \sum_{i=1}^{t} \exp\big((\mathbf{Q}\mathbf{h}_t)^\top \mathbf{K}\mathbf{h}_i\big), \qquad \mathbf{Z}_t = \sum_{i=1}^{t} \exp\big((\mathbf{Q}\mathbf{h}_t)^\top \mathbf{K}\mathbf{h}_i\big)\, \mathbf{V}\mathbf{h}_i,$$

up to the chosen approximation error. The last pointwise map can therefore form

$$\widetilde{\mathbf{h}}_t = \mathbf{h}_t + \frac{\mathbf{Z}_t}{A_t}, \qquad \mathbf{h}_t^+ = \widetilde{\mathbf{h}}_t + \mathrm{MLP}(\widetilde{\mathbf{h}}_t),$$

and clear all auxiliary coordinates. Division causes no singularity: A_t is a sum of positive exponentials and is bounded away from zero on the compact input domain.

Theorem 19 (RoPE simulation with compressed hidden dimension)**.** *Let $1 \le W \le N$, $r > 0$, and $\varepsilon > 0$. Let L be a single-head decoder transformer layer of hidden dimension d. Suppose that the simulating SWA transformer uses RoPE with a rotary pair satisfying equation 6. Then L can be approximated uniformly to error ε for every $n \le N$ and every $\mathbf{h}_1, \ldots, \mathbf{h}_n \in B^d(0, r)$ with*

$$\left\lceil \frac{N}{W} \right\rceil - 1 + W$$

RoPE–SWA layers and hidden dimension

$$2\left\lceil \frac{N}{W} \right\rceil d + d + 3 = O\left(\frac{Nd}{W}\right). \tag{23}$$

Proof. Use $p-1$ copies of the stride-shift layer from Section F.2, followed by the W accumulator layers from Section F.3. □

G Proof of the Tight Maximum Cramming Horizon

Fix the maximum sequence length $N > W\ell$. We prove Theorem 10 by adapting the shift block of Section B.4.1. The original block keeps slot zero fixed and copies the memory at $t-1$ one slot to the right. Here we give the same block stride W: it copies the memory at $t-W$, which is the leftmost position in the local attention window. A residue encoding lets one SWA layer isolate that position exactly.

Write $w = W+1$ and let $\rho(t) = t \bmod w$. The memory at position t consists of $\ell+1$ slots,

$$\mathbf{m}_t = [\mathbf{m}_t^0, \dots, \mathbf{m}_t^\ell] \in \mathbb{R}^{(\ell+1)d}.$$

We use w routed copies of this space. If $\mathbf{E}_a$ denotes the canonical injection into routed block $a \in \{0, \dots, W\}$, the full state is

$$\overline{\mathbf{m}_t} = [\mathbf{E}_{\rho(t)}\mathbf{m}_t, \mathbf{s}_t, t], \qquad \mathbf{s}_t \in \mathbb{R}^{w(\ell+1)d}. \tag{24}$$

At layer boundaries we maintain $\mathbf{s}_t = \mathbf{0}$. Initialize

$$\mathbf{m}_t^{(0)} = [\mathbf{x}_t, \mathbf{0}, \dots, \mathbf{0}]. \tag{25}$$

G.1 The stride-W shift block

Note that we could reuse the Shift Block defined in Section F.2. However, since we are not concerned by the size of the hidden dimension in this Section, we can afford to construct an exact, simpler Shift Block.

Define

$$\mathrm{Shift}_W(\mathbf{a}, \mathbf{b}) = [\mathbf{b}^0, \mathbf{a}^0, \mathbf{a}^1, \dots, \mathbf{a}^{\ell-1}], \tag{26}$$

where $\mathbf{a}$ is the memory at $t-W$ and $\mathbf{b}$ is the memory at t. This is exactly the map Shift from Section B.4.1, with $t-1$ replaced by $t-W$.

We now implement this map in one SWA layer. Set its queries and keys to zero, so attention is uniform. For $t > W$, every visible position $t-W, \dots, t$ receives weight $1/w$. Choose the value map to copy the routed memory into scratch space:

$$\mathbf{V}[\mathbf{p}, \mathbf{s}, t] = [\mathbf{0}, \mathbf{p}, 0].$$

The attention residual therefore produces

$$\widetilde{\mathbf{m}}_t = \left[\mathbf{E}_{\rho(t)}\mathbf{m}_t, \frac{1}{w}\sum_{i=t-W}^{t} \mathbf{E}_{\rho(i)}\mathbf{m}_i, t\right]. \tag{27}$$

The w consecutive positions $t-W, \dots, t$ have pairwise distinct residues modulo w. Hence the scratch block indexed by $\rho(t-W)$ contains exactly $\mathbf{m}_{t-W}/w$. If $\mathbf{P}_a$ denotes projection from routed block a, then

$$\mathbf{m}_{t-W} = w\mathbf{P}_{\rho(t-W)}\widetilde{\mathbf{s}_t}. \tag{28}$$

The pointwise component applies equation 26 to this vector and $\mathbf{m}_t$, routes the result into block $\rho(t)$, and clears the scratch space. In residual form, it is

$$\begin{aligned}\mathrm{MLP}(\widetilde{\mathbf{m}}_t) = \big[&\mathbf{E}_{\rho(t)}\,\mathrm{Shift}_W(w\mathbf{P}_{\rho(t-W)}\widetilde{\mathbf{s}_t}, \mathbf{m}_t) - \mathbf{E}_{\rho(t)}\mathbf{m}_t,\\ &-\widetilde{\mathbf{s}_t}, 0\big].\end{aligned} \tag{29}$$

As in the shift-block construction of Section B.4.1, this is a position-dependent affine map on the states under consideration and is represented exactly by the pointwise MLP construction discussed in Appendix E. Thus a single transformer layer maps

$$\mathbf{m}_t \longmapsto [\mathbf{m}_t^0, \mathbf{m}_{t-W}^0, \mathbf{m}_{t-W}^1, \dots, \mathbf{m}_{t-W}^{\ell-1}]. \tag{30}$$

G.2 Composition across layers

We claim that after $r \in \{0, \dots, \ell\}$ stride-W shift layers,

$$\mathbf{m}_t^{j,(r)} = \mathbf{x}_{t-jW} \qquad \text{for every } 0 \le j \le r \text{ and } t > rW. \tag{31}$$

For $r = 0$, this is the initialization equation 25. Suppose it holds after $r < \ell$ layers and take $t > (r+1)W$. The next shift keeps slot zero fixed, so

$$\mathbf{m}_t^{0,(r+1)} = \mathbf{x}_t.$$

For $0 \le j \le r$, it gives

$$\mathbf{m}_t^{j+1,(r+1)} = \mathbf{m}_{t-W}^{j,(r)} = \mathbf{x}_{(t-W)-jW} = \mathbf{x}_{t-(j+1)W},$$

where the induction hypothesis applies because $t - W > rW$. This proves equation 31.

Taking $r = j = \ell$ yields

$$\mathbf{m}_t^{\ell,(\ell)} = \mathbf{x}_{t-W\ell} \qquad (t > W\ell).$$

H Proofs of Section 5.2.2

Section 5.2.2 motivates the following stylized model. Under full attention, a source token $\mathbf{h}$ updates the last token in one step. Conditional on the attention weights, it adds $\alpha_s \mathbf{V}\mathbf{h}$ to the residual stream of the last token, together with the interference $\boldsymbol{\varepsilon}$ of the other tokens. Under SWA, if $\mathbf{h}$ lies k windows away, its information must be relayed k times. Write $\mathbf{h}^0 := \mathbf{h}$ and $\mathbf{h}^j \in \mathbb{R}^d$ the corresponding value that is added to the residual stream at the j-th relay. Then one relay can be modelled as

$$\mathbf{h}^{j+1} = f_j(\mathbf{h}^j) + \boldsymbol{\varepsilon}_j, \tag{32}$$

where f_j maps the value received by relay j to the value it passes on and $\boldsymbol{\varepsilon}_j$ is the interference of the other tokens in the window of relay $j+1$.

Assumption 20 (Homogeneous additive information loss)**.** *Write*

$$F_{i,j} := f_{j-1} \circ f_{j-2} \circ \cdots \circ f_i, \qquad i < j, \qquad F_{j,j} := \mathrm{id},$$

for the propagation of the source information through crossings i to $j-1$. Assume that

$$\boldsymbol{\theta}_k := F_{0,k}(\mathbf{h}^0)$$

is the intended value to be added to the residual stream (that is $\boldsymbol{\theta}_k$ is the SWA equivalent of $\alpha_s \mathbf{V}\mathbf{h}$). Then the final value added to the residual stream $\mathbf{h}^k$ differs from the intended value $\boldsymbol{\theta}_k$ as follows.

$$\mathbf{h}^k = \boldsymbol{\theta}_k + \sum_{j=1}^{k} \mathbf{e}_j, \qquad \mathbf{e}_j := F_{j,k}\big(f_{j-1}(\mathbf{h}^{j-1}) + \boldsymbol{\varepsilon}_{j-1}\big) - F_{j,k}\big(f_{j-1}(\mathbf{h}^{j-1})\big), \tag{33}$$

where $\mathbf{e}_j$ is the effect at the target of the interference injected at crossing $j-1$. We assume that $\mathbf{e}_1, \dots, \mathbf{e}_k$ are i.i.d. random vectors, independent of $\boldsymbol{\theta}_k$, with continuously differentiable density and finite, nonzero Fisher information.

We test experimentally the validity of this assumption in Appendix I. We test over one layer and show that the $\mathbf{e}_i$ are independant and gaussian. However, the scale of the gaussian varies, which explains that our prediction isn't exact in practice. We leave a more refined analysis to further work.

We use Fisher information to quantify how accurately the intended value $\boldsymbol{\theta}_k$ can be recovered from the received representation $\mathbf{h}^k$.

Definition 21 (Fisher information about the intended value)**.** For

$$\mathbf{Y}_k = \boldsymbol{\theta}_k + \sum_{j=1}^{k} \mathbf{e}_j,$$

define its scalar Fisher information by

$$J_k := \operatorname{tr} \mathbb{E}\left[\nabla_{\boldsymbol{\theta}_k} \log p(\mathbf{Y}_k \mid \boldsymbol{\theta}_k) \nabla_{\boldsymbol{\theta}_k} \log p(\mathbf{Y}_k \mid \boldsymbol{\theta}_k)^\top\right].$$

The amount of information about the intended propagated value $\boldsymbol{\theta}_k$ that is retained after k window crossings is therefore

$$J_k := J(\boldsymbol{\theta}_k + \sum_{j=1}^{k} \mathbf{e}_j).$$

The decay of J_k follows from the classical Blachman-Stam inequality.

Theorem 22 (Fisher information decay)**.** *Under Assumption 20,*

$$J_k \leq \frac{J_1}{k}.$$

If the distortions $\mathbf{e}_j$ *are Gaussian, then equality holds:*

$$J_k = \frac{J_1}{k}.$$

Proof. For independent random vectors $\mathbf{X}$ and $\mathbf{Y}$, the Blachman-Stam inequality (Stam, 1959) gives

$$\frac{1}{J(\mathbf{X}+\mathbf{Y})} \geq \frac{1}{J(\mathbf{X})} + \frac{1}{J(\mathbf{Y})}.$$

By induction, since the $\mathbf{e}_j$ are i.i.d.,

$$\frac{1}{J_k} \geq \sum_{j=1}^{k} \frac{1}{J(\mathbf{e}_j)} = \frac{k}{J_1}.$$

Therefore

$$J_k \leq \frac{J_1}{k}.$$

If the disturbances are Gaussian with common covariance $\boldsymbol{\Sigma} \succ 0$,

$$J(\sum_{j=1}^{k} \mathbf{e}_j) = \operatorname{tr}\big((k\boldsymbol{\Sigma})^{-1}\big) = \frac{J_1}{k}.$$

□

We can now obtain the total information that is retained across the receptive field.

Definition 23 (Effective information)**.** Fix a target position $t > W\ell$, so that its full receptive field is present. Write $k(s,t) := \left\lceil \frac{t-s}{W} \right\rceil$ the minimum number of SWA layers through which information from s must propagate before it can reach $t > s$. We define

$$J_{\text{eff}} := \sum_{d=0}^{W\ell-1} J(\boldsymbol{\theta}_{k(d,W\ell)} + \sum_{j=1}^{k(d,W\ell)} \mathbf{e}_j).$$

Theorem 24. *Under Assumption 20,*

$$J_{\text{eff}} \leq W H_\ell J_1, \qquad H_\ell := \sum_{k=1}^{\ell} \frac{1}{k}.$$

If the disturbances are Gaussian, then

$$J_{\text{eff}} = W H_\ell J_1.$$

Proof. Group the source positions by the number of required relays. For each $k \in \{1, \ldots, \ell\}$, exactly W positions have

$$(k-1)W < t - s \leq kW.$$

Every position in this group has $J_k \leq J_1/k$, so the group contributes at most WJ_1/k. Summing the ℓ groups gives

$$J_{\text{eff}} \leq \sum_{k=1}^{\ell} \frac{WJ_1}{k} = W H_\ell J_1.$$

Gaussian disturbances give $J_k = J_1/k$, so every inequality becomes an equality. □

We can now use the Fisher information to obtain a formula for the cramming horizon.

Assumption 25 (Cramming is proportional to transferred information)**.** *Let C denote the cramming horizon. We assume that it is proportional to the total amount of information that reaches the target:*

$$C \propto J_{\text{eff}}.$$

This is quite a natural assumption. We are simply stating that transferring twice as much information allows the model to encode a sentence twice as long.

Theorem 26 (Harmonic cramming horizon)**.** *Under Assumptions 20 and 25, the cramming horizon C of an ℓ layers SWA transformer with window size W satisfies*

$$C \leq W H_\ell.$$

If the disturbances are Gaussian, then

$$\boxed{C = W H_\ell}.$$

Proof. First consider the one layer case. Then every one of the W source positions reaches the target with Fisher information J_1, so

$$J_{\text{eff}}^{\text{1layer}} = W J_1.$$

Theorem 10 shows that the corresponding cramming horizon is W. This fixes the proportionality in Assumption 25:

$$C = \frac{J_{\text{eff}}}{J_1}.$$

Theorem 24 gives $J_{\text{eff}}/J_1 \leq W H_\ell$, with equality for Gaussian disturbances. □

Remark 27. The validity of Assumption 20 can be discussed. Note howver that if we only consider attention without the MLP component, it can be replaced by the much more natural following assumption.

Let $\mathbf{h}^0 \in \mathbb{R}^d$ be the source representation and write

$$\mathbf{B}_{i,j} := \mathbf{A}_{j-1}\mathbf{A}_{j-2}\cdots\mathbf{A}_i, \qquad i < j.$$

After k window crossings,

$$\mathbf{h}^k = \mathbf{B}_{0,k}\mathbf{h}^0 + \sum_{j=0}^{k-1} \mathbf{B}_{j+1,k}\boldsymbol{\varepsilon}_j.$$

Write

$$\boldsymbol{\theta}_k := \mathbf{B}_{0,k}\mathbf{h}^0, \qquad \mathbf{e}_j := \mathbf{B}_{j,k}\boldsymbol{\varepsilon}_{j-1}, \quad j = 1, \ldots, k.$$

We assume that $\mathbf{e}_1, \ldots, \mathbf{e}_k$ are i.i.d. random vectors with continuously differentiable density and finite, nonzero Fisher information. Thus

$$\mathbf{h}^k = \boldsymbol{\theta}_k + \sum_{j=1}^{k} \mathbf{e}_j.$$

I Empirical Support for Assumption 20

Theorems 22 and 26 hold with equality when the distortions $\mathbf{e}_j$ are Gaussian and independent. Appendix I.1 measures them in a pretrained model, for a fixed source and a random context, and finds them independent and Gaussian up to a scale that varies across contexts, which costs a factor $\kappa \in [1.05, 1.21]$ in Fisher information. Appendix I.2 shows that this scale lowers the predicted horizon $W H_\ell$ by 2–9% for $\ell = 24$, uniformly in W.

I.1 GAUSSIANITY AND INDEPENDENCE OF THE DISTORTIONS

Setting. Fix W and an L-layer transformer run under the causal SWA mask of size W: position p attends to $p-W,\dots,p$. Write $\mathbf{h}_p^{(r)}$ for the residual stream of position p after layer r.

(i) *Sequence.* $\mathbf{x}=(x_0,\boldsymbol{\sigma},\mathbf{f},\boldsymbol{\epsilon})$: a fixed first token, a fixed source phrase of n_s tokens, a filler of n_f random tokens, a fixed end phrase of n_e tokens. The source $s\le n_s$ is the position of the token of $\boldsymbol{\sigma}$ that carries the information (*boy* in "I am a boy."); the final position is $T-1$. The filler is the only random element, so the states of $x_0,\dots,x_s$ are the same for every sample.

(ii) *Range.* $\mathbf{h}_{T-1}^{(r)}$ depends on x_s only if $r\ge k-1$, with $k:=\lceil (T-1-s)/W\rceil$ the number of crossings. We take $T-1-s=W$ for $k=1$ and $T-1-s=W+\lceil W/2\rceil$ for $k=2$.

(iii) *First crossing.* In layer 0, for every position p with $s<p\le s+W$ and every head, the attention update $\sum_i \alpha_{p,i}\mathbf{v}_i$ is replaced by its source term $\alpha_{p,s}\mathbf{v}_s$. Everything else is unchanged. Write $\tilde{\mathbf{h}}$ for the result.

(iv) *Last crossing* ($k=2$). In layer 1, for the final position and every head, $\sum_i \alpha_{T-1,i}\mathbf{v}_i$ is replaced by $\sum_{i\in R}\alpha_{T-1,i}\mathbf{v}_i$, where $R=\{i:\ T-1-W\le i\le s+W\}$ are the positions of its window that saw the source in layer 0. Write $\hat{\mathbf{h}}$ for the result.

(v) *Distortions*, read at the final position after layer $k-1$:

$$\mathbf{e}_1:=\mathbf{h}_{T-1}^{(k-1)}-\tilde{\mathbf{h}}_{T-1}^{(k-1)},\qquad \mathbf{e}_2:=\mathbf{h}_{T-1}^{(1)}-\hat{\mathbf{h}}_{T-1}^{(1)}.$$

Step (iii) removes the interference $\boldsymbol{\varepsilon}_0$ of Equation 32, step (iv) does the same for $\boldsymbol{\varepsilon}_1$. Hence $\mathbf{e}_1$ and $\mathbf{e}_2$ are the distortions of Assumption 20 for 1 and 2 crossings respectively. We measure $\{(\mathbf{e}_1,\mathbf{e}_2)(\mathbf{f})\}$ over independent fillers $\mathbf{f}$.

Example. $W=4$, $\boldsymbol{\sigma}=$ "I am a boy." at positions 1–5, $s=4$, a filler at 6–8, an end phrase at 9–11; so $k=2$. In layer 0 the readers $5,\dots,8$ get $\alpha_{p,4}\mathbf{v}_4$ alone under (iii). In layer 1 position 11 attends to $7,\dots,11$, of which $R=\{7,8\}$ saw the source; under (iv) it keeps $\alpha_{11,7}\mathbf{v}_7+\alpha_{11,8}\mathbf{v}_8$ alone. $\mathbf{e}_1$ is the effect of the terms $\alpha_{p,i}\mathbf{v}_i$, $i\neq 4$, of positions 5–8 in layer 0, after the MLP of layer 0 and layer 1; $\mathbf{e}_2$ that of the terms $\alpha_{11,i}\mathbf{v}_i$, $i\in\{9,10,11\}$, after the MLP of layer 1.

Protocol. Qwen2.5-0.5B ($L=24$, $d=896$), $W\in\{16,32,64\}$, $k\in\{1,2\}$, x_0 a newline, the six sources of Table 3. Filler tokens are words drawn uniformly from the 20,000 most frequent English words, tokenized and cut to n_f tokens. $N=2{,}048$ fillers per cell (source, W, k), 1,024 for $W=64$; 18 cells per row of Tables 4 and 5, reported as mean $\pm$ standard deviation over cells.

Statistics. Each sample batch $\mathbf{e}_i$ is split in two halves. The mean, covariance and $m=\min(256,N/4)$ leading principal directions are fitted on the first half; the second half is whitened there into $\mathbf{z}\in\mathbb{R}^m$. If $\mathbf{e}$ is Gaussian, then the whitening produces a standard normal variable: $\mathbf{z}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_m)$. Table 4 reports the median norm of $\mathbf{e}$ relative to $\mathbf{h}_{T-1}^{(k-1)}$; the coefficient of variation of $\|\mathbf{z}\|^2$ (Gaussian: $\sqrt{2/m}$); the Fisher excess $\kappa:=\mathbb{E}\|\mathbf{z}\|^2\,\mathbb{E}\|\mathbf{z}\|^{-2}\ge 1$, the ratio of the Fisher information of a Gaussian scale mixture $\sigma\mathbf{g}$ with known scale to that of the Gaussian with the same covariance, equal to 1 iff σ is constant; the median excess kurtosis γ_2 of 256 random unit projections of $\mathbf{z}$, before and after dividing every sample by its norm (sphere); and Δ_{24}, the relative drop of WH_{24} computed from the empirical law of $\|\mathbf{z}\|^2$ as in Appendix I.2. Table 5 tests $\mathbf{e}_1\perp\mathbf{e}_2$ at $k=2$: the Spearman correlation $\rho_{\|\cdot\|}$ of $\log\|\mathbf{e}_1\|$ and $\log\|\mathbf{e}_2\|$; the mean squared cross-correlation $\rho^2:=\mathrm{tr}(\mathbf{C}_A\mathbf{C}_B^\top)/m$ of the whitened $\mathbf{z}_1,\mathbf{z}_2$, with $\mathbf{C}_A,\mathbf{C}_B$ their cross-covariances on the two halves (unbiased, 0 under independence, 1 if $\mathbf{z}_2=\mathbf{z}_1$); and the top canonical correlation ρ_{cca} with directions fitted on one half and evaluated on the other. Every statistic is also computed on Gaussian samples with the same moments and size (independent pairs for Table 5), processed identically; its mean over cells is given in parentheses.

I.2 EFFECT OF A NON-CONSTANT SCALE ON WH_ℓ

Table 4 shows distortions that are Gaussian in direction but whose scale varies across contexts. Model them as $\mathbf{e}_j=\sigma_j\mathbf{g}_j$ with $\mathbf{g}_j$ i.i.d. Gaussian and σ_j i.i.d. scales, independent of the $\mathbf{g}_j$. Two

Table 3: **The six fixed sources.** The source token x_s is fixed by construction.

source phrase $\boldsymbol{\sigma}$	x_s	end phrase $\boldsymbol{\epsilon}$
I am a boy.	boy	When people talk about me they say that
My name is Alice.	Alice	Yesterday my friend called me and said hello
It was raining.	raining	I looked out of the window and took my
I was born in France.	France	The language I speak at home is
My car is red.	red	In the parking lot I always look for something
She is a doctor.	doctor	Every day she goes to work at the

Table 4: **Both distortions are Gaussian in direction and close to Gaussian in scale: $\kappa \leq 1.21$, at most 9% off the horizon.** Mean $\pm$ sd over 18 cells (6 sources, $W \in \{16, 32, 64\}$); mean of the Gaussian reference in parentheses. j: crossing.

k	j	$\|\mathbf{e}_j\|/\|\mathbf{h}\|$	CV	κ	γ_2	γ_2 (sphere)	Δ_{24} (%)
1	1	0.840 ± 0.164	0.33 ± 0.15 (0.10)	1.094 ± 0.070 (1.010)	0.28 ± 0.28 (-0.02)	-0.050 ± 0.012 (-0.054)	4.5 ± 3.0 (0.6)
2	1	0.058 ± 0.017	0.69 ± 0.30 (0.10)	1.209 ± 0.079 (1.011)	0.97 ± 0.77 (-0.03)	-0.051 ± 0.013 (-0.056)	8.5 ± 2.1 (0.6)
2	2	0.514 ± 0.083	0.24 ± 0.08 (0.10)	1.046 ± 0.020 (1.011)	0.10 ± 0.09 (-0.02)	-0.053 ± 0.016 (-0.048)	2.4 ± 0.9 (0.6)

Table 5: **The two distortions are independent in direction; their scales are weakly and negatively correlated.** $k = 2$, mean $\pm$ sd over 18 cells; mean of the independent Gaussian reference in parentheses.

$\|\mathbf{e}_2\|/\|\mathbf{e}_1\|$	$\rho_{\|\cdot\|}$	ρ^2	ρ_{cca}
9.7 ± 3.6	-0.174 ± 0.141 (0.005)	-0.0018 ± 0.0033 (-0.0013)	-0.013 ± 0.047 (0.016)

cases. If the scale is common to all crossings, $\sigma_j = \sigma$, then $\sum_{j \leq k} \mathbf{e}_j = \sigma \sum_j \mathbf{g}_j$ and σ cancels in J_k/J_1: Theorems 22 and 26 hold unchanged. If the scales are independent (the worst case), then conditionally on the scales $\sum_{j \leq k} \mathbf{e}_j$ is Gaussian with covariance $S_k^2 \boldsymbol{\Sigma}$, $S_k^2 := \sum_{j \leq k} \sigma_j^2$, so that $J_k \propto \mathbb{E}[S_k^{-2}]$ and the factor $1/k$ of Theorem 22 becomes

$$r_k := \frac{\mathbb{E}[S_k^{-2}]}{\mathbb{E}[\sigma^{-2}]} \leq \frac{1}{k},$$

by the inequality between arithmetic and harmonic means, with equality iff σ is constant. The horizon WH_ℓ becomes $W \sum_{k \leq \ell} r_k = WH_\ell(1 - \Delta_\ell)$ with $\Delta_\ell \geq 0$: a non-constant scale can only lower it, by the same factor for every W. To second order in $c^2 := \mathrm{Var}(\sigma^2)/\mathbb{E}[\sigma^2]^2$, $r_k \approx \frac{1}{k} \frac{1+c^2/k}{1+c^2}$ and $\kappa \approx 1 + c^2$, hence

$$\Delta_\ell \approx \Big(1 - \frac{1}{\kappa}\Big)\Big(1 - \frac{H_\ell^{(2)}}{H_\ell}\Big), \qquad H_\ell^{(2)} := \sum_{k \leq \ell} \frac{1}{k^2},$$

with $1 - H_\ell^{(2)}/H_\ell = 0.50,\ 0.58,\ 0.61,\ 0.66$ for $\ell = 12, 24, 36, 64$, and $\Delta_\ell \to 1 - 1/\kappa$ as $\ell \to \infty$. With the measured $\kappa = 1.09,\ 1.21,\ 1.05$ of the three rows of Table 4, the formula gives $\Delta_{24} = 4.9\%,\ 9.9\%,\ 2.5\%$. This formula is confirmed by the Monte Carlo values of the table, computed from the empirical law of $\|\mathbf{z}\|^2$ without the expansion, which are $4.5\%,\ 8.5\%,\ 2.4\%$. A negative correlation between the scales of successive crossings, as in Table 5, reduces $\mathrm{Var}(S_k^2)$ and hence Δ_ℓ further. At $W = 16$ the prediction $H_{24}W = 60.4$ tokens becomes at least 55, within the 10-token resolution of Section 6.2. The dependence on W stays linear and the dependence on ℓ stays harmonic up to the slowly varying factor $1 - \Delta_\ell$.

J SUPPORTING EXPERIMENTS

Appendix J.1 replicates the cramming horizon experiment of Section 6.2 on Qwen3-1.7B. Appendix J.2 checks that information never propagates beyond the hard limit $W\ell$. Appendix J.3 lists the architectural differences between the models used. Appendix J.4 isolates the effect of the number of layers on the cramming horizon.

J.1 THE PLATEAU ALSO HOLDS FOR QWEN3-1.7B

Section 6.2 uses Qwen2.5-0.5B. We repeat the same sweep on Qwen3-1.7B, which has 28 decoder layers, varying M from 1 to 1000.

Figure 6 has the same shape as Figure 4. The curves first rise, then flatten near 70, 140–150, and 280–300 tokens for $W = 16, 32, 64$. The predictions are $H_{28}W = 62.8,\ 125.6,\ 251.4$. The coefficient is slightly larger than on Qwen2.5-0.5B, but the saturation in M and the linear dependence on W both replicate.

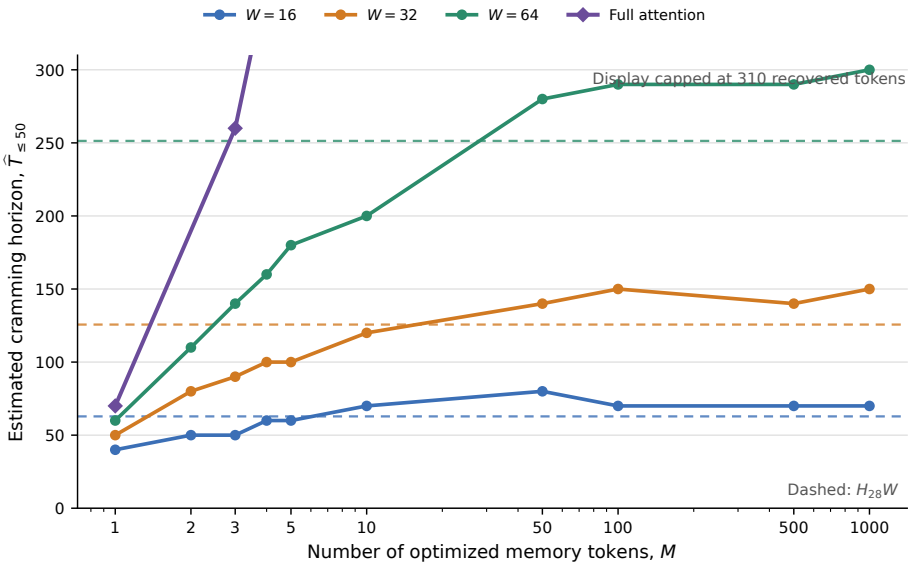


Figure 6: **The plateau replicates on Qwen3-1.7B.** Solid curves show the recoverable output length as the optimized soft prompt grows; dashed lines show $H_{28}W$. Full attention (purple) is included for comparison.

J.2 INFORMATION NEVER PROPAGATES BEYOND $W\ell$ TOKEN POSITIONS

We check experimentally that information cannot propagate beyond the hard limit $W\ell$ in Figure 7. For a given token, we compute the gradient of its output with respect to every earlier token and identify the furthest token with a nonzero gradient. We test both a fixed window size, and varying windows that change from layer to layer. We also check that full attention does not have this limitation.

J.3 ARCHITECTURE OF THE MODELS USED IN OUR EXPERIMENTS

Table 6 records the architectural differences between models.

Table 6: **Architecture of the models used in our experiments.** H_Q/H_{KV} is the query/key-value head count and d_h is the head dimension.

Model	Params.	ℓ	d_{model}	d_{FFN}	H_Q/H_{KV}	d_h	$\lvert V\rvert$	Residual norm	Q/K norm	Tied emb.	RoPE	Tested W
Qwen2.5-0.5B	0.5B	24	896	4,864	14/2	64	151,936	RMS, $\epsilon = 10^{-6}$	no	yes	$\theta = 10^6$	16, 32, 64
Qwen3-0.6B	0.6B	28	1,024	3,072	16/8	128	151,936	RMS, $\epsilon = 10^{-6}$	RMS	yes	$\theta = 10^6$	16, 32, 64
Qwen3-1.7B	1.7B	28	2,048	6,144	16/8	128	151,936	RMS, $\epsilon = 10^{-6}$	RMS	yes	$\theta = 10^6$	16, 32, 64
Qwen3-4B	4.0B	36	2,560	9,728	32/8	128	151,936	RMS, $\epsilon = 10^{-6}$	RMS	yes	$\theta = 10^6$	16, 32, 64
Qwen3-8B	8.0B	36	4,096	12,288	32/8	128	151,936	RMS, $\epsilon = 10^{-6}$	RMS	no	$\theta = 10^6$	16, 32, 64
Qwen3-32B	32B	64	5,120	25,600	64/8	128	151,936	RMS, $\epsilon = 10^{-6}$	RMS	no	$\theta = 10^6$	16, 32, 64
Gemma-2-2B	2.6B	26	2,304	18,432	8/4	256	256,128	RMS, $\epsilon = 10^{-6}$	RMS	yes	$\theta = 10^4$	16, 32, 64
Gemma-2-9B	9.2B	42	3,584	28,672	16/8	256	256,128	RMS, $\epsilon = 10^{-6}$	RMS	yes	$\theta = 10^4$	16, 32, 64

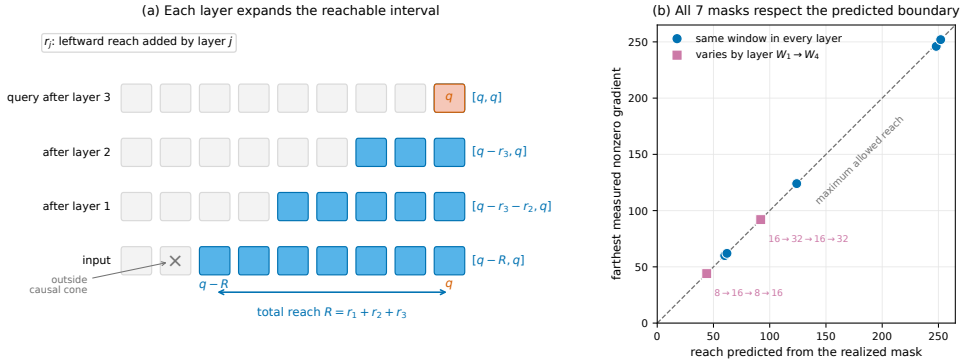


Figure 7: **Information never travels further than the theory allows.** The gradients reach the predicted boundary exactly and vanish one position beyond it.

J.4 Depth alone reproduces the harmonic factor

The models in Table 1 differ in weights, size, and architecture, not only in depth. To vary depth alone, we fix $W = 64$, keep only the first ℓ decoder layers of a checkpoint, and repeat the horizon search. The embedding, final normalization, and output head are untouched. For Gemma-2-9B we report $\ell \geq 26$ only, since shallower settings gave no two-sided 50% crossing on our target-length grid.

Table 7 compares the measured horizons with WH_ℓ. Qwen3-32B tracks the prediction closely: midpoint 175 versus $H_8 W = 173.9$, and 215 versus $H_{16} W = 216.4$. $\ell = 32$ gives the 230–240 bracket, while the reference is 259.7. Gemma-2-9B holds a stable ratio between 0.83 and 0.87 across 26, 32, and 42 layers.

Table 7: **Depth alone moves the cramming horizon along WH_ℓ.** All conditions use $W = 64$ and $M = 200$. Each observed bracket is the adjacent target-length pair straddling 50% exact recovery.

Model	Active layers ℓ	Observed bracket	Midpoint	WH_ℓ	Observed/predicted
Qwen3-32B	4	120–130	125	133.3	0.937
Qwen3-32B	8	170–180	175	173.9	1.006
Qwen3-32B	16	210–220	215	216.4	0.994
Qwen3-32B	32	230–240	235	259.7	0.905
Gemma-2-9B	26	200–210	205	246.7	0.831
Gemma-2-9B	32	220–230	225	259.7	0.866
Gemma-2-9B	42	230–240	235	276.9	0.849

J.5 Sensitivity to the empirical recovery cutoff

The main experiments define the empirical cramming horizon using a 50% exact-recovery threshold. We repeat a targeted subset using a lower 20% threshold to test whether the comparison with WH_ℓ depends strongly on this convention. We use each checkpoint at its full depth with $W = 64$ and $M = 200$. The model remains frozen, and the soft prompt is optimized for up to 5,000 steps. We evaluate target lengths in 10-token increments and confirm both endpoints of each crossing using 100 trials.

Lowering the cutoff moves the estimated horizon by only 10–20 tokens across these checkpoints. The normalized midpoint changes by at most 0.07. Qwen3-0.6B and Qwen3-4B remain close to WH_ℓ, while Gemma-2-9B and Qwen3-32B remain below it. The qualitative comparison with the theoretical scale is therefore stable under this change in the empirical cutoff.

Table 8: **Sensitivity of the estimated cramming horizon to the recovery cutoff.** Parentheses report the number of exactly recovered targets out of 100 at the lower and upper endpoints. The final column reports the ratio between the bracket midpoint and WH_ℓ under the 50% and 20% definitions.

Model	Layers ℓ	WH_ℓ	50% bracket	20% bracket	Midpoint$/WH_\ell$
Qwen3-0.6B	28	251.4	260–270 (76/100, 42/100)	270–280 (42/100, 7/100)	$1.05 \rightarrow 1.09$
Qwen3-4B	36	267.2	260–270 (56/100, 27/100)	270–280 (30/100, 15/100)	$0.99 \rightarrow 1.03$
Gemma-2-9B	42	276.9	230–240 (71/100, 42/100)	250–260 (23/100, 12/100)	$0.85 \rightarrow 0.92$
Qwen3-32B	64	303.6	230–240 (62/100, 44/100)	240–250 (42/100, 17/100)	$0.77 \rightarrow 0.81$