COPYING UNDER REPETITION: FROM PREFIX MATCHING TO POSITION HASHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Frontier LLMs that copy random strings accurately can overrun or truncate long blocks of repeated characters. Repetition removes the local content cue that induction-style copying relies on, and what is left to resolve—which occurrence to continue from, and when to stop—is positional. Position can reach the output along two routes: through the attention weights a head places on value vectors it leaves unchanged, or through offsets written into those value vectors, which a later layer can read and retrieve with. We study this distinction in *horizon-uniform copying*, which requires one transformer to copy every string of length $L \leq N$ without parameters specialized to the realized length. Our value-side construction, *VO-rotation* with exact access to modular residues, copies every string of length $L \leq N$ for any fixed finite vocabulary and all sufficiently large N, using two layers, $\Theta(\log N / \log \log N)$ prime-period heads, and $O(\log \log N)$-bit precision. The same retrieval rule selects the next source token and the end marker, handling arbitrary repetition and termination together. We also prove a width-independent saturation bound: in our finite-precision model, transformers whose only positional input is an eventually-constant additive attention bias fail on all-identical sources once their saturation threshold is $o(N)$. Across three training seeds, learned VO models copy beyond the training limit, with an accuracy transition near the product of their supplied periods, consistent with learned use of the modular representation. In contrast, standard positional encoding baselines lose accuracy under repetition near the training limit.

1 INTRODUCTION

Autoregressively copying text is a fundamental primitive for modern large language models (LLMs). LLMs copy when they reuse retrieved documents, quote code or data, or transfer a specific answer from a chain of thought into the final response. Historically, this copying task has motivated explicit pointer or copy mechanisms in sequence models (Vinyals et al., 2015; Gu et al., 2016; See et al., 2017). Meanwhile, transformers seem to learn to copy easily, often via the "induction head" mechanism (Olsson et al., 2022): they match the already-copied prefix to its earlier occurrence in the prompt and retrieve the subsequent token. For random sequences, short local contexts uniquely identify the occurrence. Jelassi et al. (2024) formalized this content-addressed mechanism, showing that a two-layer transformer (with a particular position encoding) can copy strings of length $\leq L$ with $O(\log \log L)$ precision—provided the input contains no repeated n-grams.

Surprisingly, however, removing this non-repetition assumption seems to change the difficulty of copying qualitatively. As shown in Figure 1, frontier LLMs that copy random strings near-perfectly will sometimes fail catastrophically on strings with long repeated runs of letters, either looping indefinitely or stopping early. Repetition changes what the task demands: when many positions carry the same local content, matching the already-copied prefix no longer picks out which occurrence to continue from. Reliable copying must therefore resolve occurrences that local content alone cannot distinguish.

To make this problem concrete, consider one step of autoregressive copying, with current context

$$\langle \text{BOS} \rangle,\ s_1, \ldots, s_L,\ \langle \text{EOI} \rangle^{1},\ \langle \text{COPY} \rangle,\ s_1, \ldots, s_k,$$

[1] $\langle \text{EOI} \rangle$ stands for "end of input" and $\langle \text{EOS} \rangle$ for "end of sentence".

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

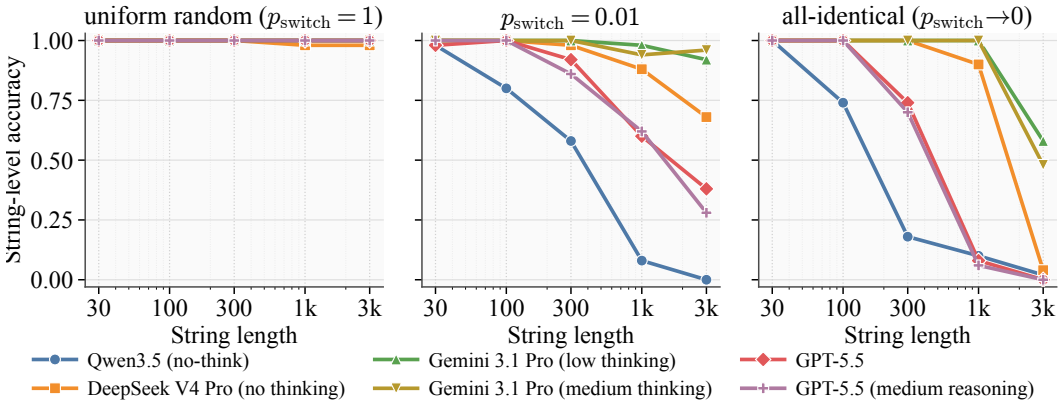


Figure 1: LLM exact-copy diagnostic on lowercase strings. For a fixed switch probability p_{switch}, a source string is generated by concatenating latent constant-letter runs: each run chooses a letter uniformly from the 26 lowercase English letters and has length $R \sim \text{Geom}(p_{\text{switch}})$. The three plotted distributions are uniform-random strings, $p_{\text{switch}} = 1$; repeated strings, $p_{\text{switch}} \approx 0.01$; and all-identical strings, the limit $p_{\text{switch}} \to 0$. Adjacent latent runs may choose the same letter, so observed same-letter stretches can be longer than one run. Experimental details are in Appendix A.

for some $k \leq L$. The task is to output s_{k+1} if $k < L$, or output $\langle \text{EOS} \rangle$ if $k = L$. The all-identical case, where $s_i = \sigma$ for all i, isolates the termination problem plaguing commercial LLMs. Continuing the copy requires only emitting another σ, but deciding when to emit $\langle \text{EOS} \rangle$ requires tracking the length of the copied segment. Local content windows lying inside the repeated run look the same, thus they cannot distinguish the termination step from the previous step which demands continuation.

Why does repetition remain difficult for models that already use positional encodings (PEs)? We investigate how the way position enters attention affects the model's ability to use it for retrieval and termination. We distinguish two pathways through the attention computation. *Score-side* PEs, including RoPE (Su et al., 2024) and ALiBi (Press et al., 2022), inject position into the attention logits: they influence the output only through the attention weights they place on value vectors which they leave unchanged. *Value-side* PEs can instead transform value vectors by relative position, so the representations being aggregated themselves depend on the offsets attended to. On an all-identical source, a first-layer score-side head receives the same value vector from every source token, even if it assigns different weights to different occurrences. The only effective position signal passed to later layers is through varying weights to the delimiter tokens (e.g. BOS). Thus selecting different source occurrences does not directly return different value content: positional distinctions must be recovered from the mixing weights. A value-side PE can instead make the returned value depend on the selected offset, giving later layers a representation they can use for retrieval. This contrast leads us to study how positional representations support reliable copying, both as an expressivity question and as a learning problem.

1.1 Our Contributions

We formalize the expressivity question in a *horizon-uniform* framework: one model must copy every source of length $L \leq N$ with the same parameters. Its width, depth, and precision may grow with the design horizon N, but its parameters cannot specialize to the realized length. This captures a natural requirement for a model designed for a finite context window, and lies between length-specific circuit expressivity (Li et al., 2024; Merrill & Sabharwal, 2023) and fixed-resource expressivity over unbounded lengths (Yang et al., 2025; 2026). The next-token copying map itself has low circuit complexity: it admits DLOGTIME-uniform AC^0 circuits of constant depth and linear size, with positional routing encoded in the wiring (Appendix C). Our framework asks what positional representations and computational resources a transformer needs to implement this routing reliably across source lengths.

Within this framework, we prove two concrete cases of the score-side/value-side contrast.

A width-independent copying obstruction. On the score side, we prove a width-independent impossibility result for *eventually-constant bias-only* (ECB) transformers, whose only positional information enters through a pre-softmax additive attention bias that becomes constant beyond some distance. The analysis covers NoPE (Kazemnejad et al., 2023), recursive ALiBi (Press et al., 2022), and bounded-window Hard-ALiBi (Jelassi et al., 2024) within this class. The proof tracks identical-token runs after an arbitrary fixed prefix, layer by layer: beyond a quantitative length threshold, appending another identical token no longer changes the next-token logits. Applied to copying, a model that emits the required repeated letters then loops indefinitely instead of stopping. Increasing width does not remove this obstruction: for the instances in Corollary 3.5, models of depth $D(N)$ and precision $p(N)$ fail whenever $D(N)2^{p(N)} = o(N)$. This includes depth $D = N^{0.01}$ and precision $p \leq 0.9 \log_2 N$.

Copying and termination with VO-rotation. On the value side, we construct a two-layer transformer that copies strings with arbitrary repetition and terminates correctly. The construction uses *VO-rotation*, a natural value-side counterpart of RoPE (Su et al., 2024) that rotates values rather than queries and keys, and with a prime-number period per head instead of the geometrically-spaced frequencies in RoPE. Layer 1 selects an anchor ($\langle$BOS$\rangle$ in the source segment, $\langle$COPY$\rangle$ in the copy segment) and writes the query's offset into the value stream. Layer 2 uses these offsets to retrieve the next source token or the end marker, which the readout maps to either the next input token or $\langle$EOS$\rangle$: copying and termination share one retrieval rule. A head of period q identifies the offset only modulo q; by the Chinese Remainder Theorem, $\Theta(\log N / \log \log N)$ prime-period heads distinguish offsets within the horizon. The chosen small periods keep each head's phases distinguishable at low precision: with phase indices generated modulo these primes, the construction copies every string of length $L \leq N$ at precision $O(\log \log N)$ (Section 3.3).

Learning to copy under repetition. In synthetic experiments, standard positional baselines lose accuracy under repetition near the training limit, while a shallower, narrower VO model learns to copy across all three evaluated distributions (Section 4). Across three training seeds, VO copies perfectly at all evaluated lengths within the training range and maintains high average accuracy through length 5000, near the period product $5 \cdot 7 \cdot 11 \cdot 13 = 5005$. This pattern supports learned use of the modular position representation (Figure 4).

2 Related Work

Copying under repetition requires distinguishing occurrences with the same content. Jelassi et al. (2024) construct a two-layer transformer that copies exponentially long strings by content-based n-gram hashing, assuming no repeated n-grams; their copying error measures the first L output tokens. Our two-layer construction retrieves by anchor-relative position, allowing arbitrary repetition and including correct termination in the guarantee. Repetition also distinguishes easy and difficult copying tasks in studies of length generalization (Zhou et al., 2024). Huang et al. (2025) prove that Limit Transformers satisfying their Periodic and Local conditions cannot copy arbitrary strings. Our ECB lower bound instead quantifies layerwise saturation under finite-precision execution, independently of width.

Positional encodings differ both in where they enter the computation and in how they represent position. Value-side representations include the additive relative embeddings of Shaw et al. (2018) and the geometric transformations of GTA (Miyato et al., 2024). RoVE (García-Castellanos et al., 2026) studies relative value rotations through an attentive-convolution interpretation and language-modeling experiments. ConveRT (Henderson et al., 2020) uses learned absolute-position embeddings indexed by two coprime moduli. Our focus is on prime-period value rotations that encode anchor-relative offsets as modular residues, with explicit precision and head-count bounds for reliable copying (Theorem 3.8).

Expressivity results depend on which resources may grow with input length. Circuit complexity bounds characterize transformer computation (Merrill & Sabharwal, 2023; Strobl et al., 2024), but the fixed-context next-token copy map already lies in DLOGTIME-uniform AC^0 (Section C). Temporal-logic analyses characterize fixed-precision NoPE transformers and the effect of local attention (Li & Cotterell, 2025; 2026), while length-generalization theory studies when bounded-length training suffices for unbounded-length correctness (Chen et al., 2025; Yang et al., 2026). Our horizon-uniform

framework asks how positional and computational resources must scale with a finite design horizon, requiring the same parameters to work across all source lengths within it.

Sequential depth and explicit identifiers provide alternative routes to copying. Fan et al. (2025) implement copying by iterating a fixed block for n steps, and Kraus et al. (2026) use a growing signpost alphabet to address copying and retrieval barriers in chain-of-thought reasoning. Our construction uses a fixed vocabulary and two layers, without intermediate reasoning or indexing tokens.

3 Theoretical Framework

Under repetition, content matching need not distinguish the occurrence to copy next. We study two ways positional information can reach the output: through attention weights over value vectors, or through position-dependent transformations of the values themselves. The results establish failure for eventually-constant bias-only models under a specified finite-precision execution model, and sufficiency of a modular value-side construction for two-layer copying. Appendix D gives the full execution model, architecture definitions, and proofs.

3.1 Horizon-Uniform Copying

Fix a design horizon $N \in \mathbb{N}$. Let $\Gamma = \mathcal{V} \cup \{\langle \text{BOS} \rangle, \langle \text{EOI} \rangle, \langle \text{COPY} \rangle, \langle \text{EOS} \rangle\}$ be the token alphabet for the copy task. For a transformer T, write g_T for its deterministic greedy next-token map, under a fixed tie-breaking rule, on contexts in its *model domain*, i.e., the contexts that fit within its context window. We consider transformers T_N with context window at least $2N + 3$. For any source length $L \le N$, the input prompt has the form

$$P(s) := \underbrace{\langle \text{BOS} \rangle,\ s_1, \ldots, s_L,\ \langle \text{EOI} \rangle}_{\text{source segment}},\ \langle \text{COPY} \rangle,$$

where $s_1, \ldots, s_L \in \mathcal{V}$ and there is no restriction on repetition. After seeing $\langle \text{COPY} \rangle$, the model must autoregressively output $s_1, \ldots, s_L, \langle \text{EOS} \rangle$.

Definition 3.1 (Horizon-uniform copying). *Fix a design horizon N. A model T_N solves $\text{Copy}_{\le N}$ if the same parameters, execution convention, and readout satisfy the following condition for every source string $s \in \mathcal{V}^L$ with $L \le N$. Starting from the prompt $P(s)$, recursively define*

$$y_r = g_{T_N}\big(P(s), y_1, \ldots, y_{r-1}\big), \qquad 1 \le r \le L + 1.$$

Then

$$(y_1, \ldots, y_L, y_{L+1}) = (s_1, \ldots, s_L, \langle \text{EOS} \rangle).$$

A family of models $\{T_N\}$ solves horizon-uniform copying if each T_N solves $\text{Copy}_{\le N}$.

Finite precision. Both our results assume deterministic finite-precision floating-point execution, as detailed in Appendix D.1. Throughout this section, $p = p(N)$ is a resource that may scale with the design horizon N. We follow the iterative-rounding and left-to-right summation convention of Li et al. (2024, Definition 3.1) for the softmax numerator and denominator, although tie-breaking and overflow conventions differ. This execution model lets us track how repeated contributions lose length information through rounding, and how value-side offsets remain usable under the same numerical arithmetic.

The proofs rely on the following two facts about finite-precision arithmetic:

- When intermediates remain finite, repeated addition and rounding of an identical summand reaches a fixed point after at most 2^p states: $a, \mathsf{fl}(a + b), \mathsf{fl}(\mathsf{fl}(a + b) + b), \ldots$ is eventually constant.
- With the max-shifted softmax convention, sufficiently large post-shift logit gaps yield *one-hot attention*, which is used in our VO construction.

3.2 Lower Bound for Eventually-Constant Bias-Only Transformers

Definition 3.2 (Eventually-constant bias-only transformer class). *A family $\{T_N\}_{N \in \mathbb{N}}$ is in $\mathcal{T}_{\text{ECB}}$ if, for each horizon N, T_N is a depth-$D(N)$ causal transformer with arbitrary number of attention*

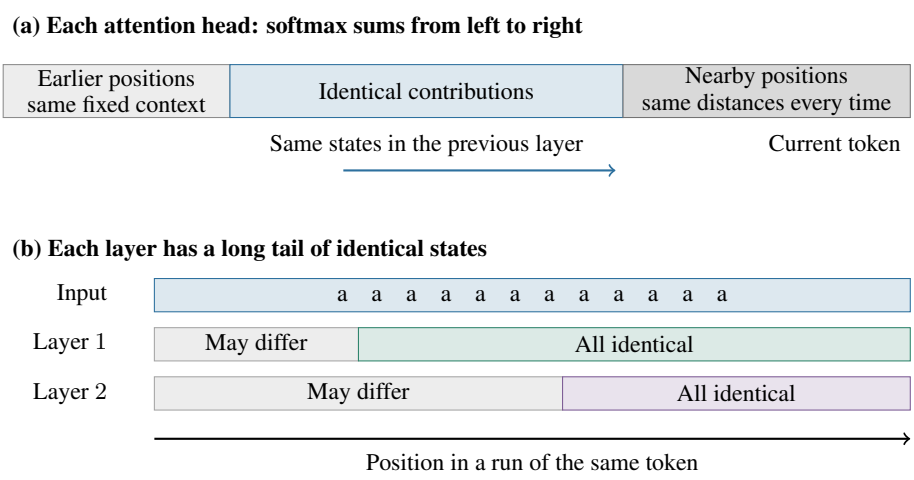


Figure 2: How ECB transformers lose track of repetition through rounded accumulation. Colored tails indicate guaranteed identical states within each layer; gray states may differ (schematic).

heads and hidden dimensions, using deterministic $p(N)$-bit execution. Its only positional dependence is a distance-dependent additive bias on the attention logits. For head h in layer ℓ, the attention logit from query position i to key position j is defined as:

$$e_{i,j}^{(\ell,h)} = \mathsf{fl}\big((q_i^{(\ell,h)})^\top k_j^{(\ell,h)} - b_h^{(\ell)}(i-j)\big),$$

where each $b_h^{(\ell)}$ is an eventually constant function: for some $n_h^{(\ell)}$, $b_h^{(\ell)}(d+1) = b_h^{(\ell)}(d)$ for all $d \geq n_h^{(\ell)}$. Here ECB stands for eventually-constant bias-only. *The full formal definition is in Definition D.5.*

Write (x, σ^k) for a fixed prefix x followed by k copies of σ. The proof tracks where the hidden states become identical within each layer. Write $\bar{n}_\ell = \max_h n_h^{(\ell)}$ and define $\kappa_0 = 0$, $\kappa_\ell = \kappa_{\ell-1} + \bar{n}_\ell + 2^p$. Once the previous layer has identical states beyond $\kappa_{\ell-1}$, a sufficiently late query sees a fixed past, a block of identical softmax addends, and a near window with a fixed list of relative distances (Figure 2). The bias is constant on the far positions, and at least 2^p middle-block additions make the accumulator independent of further repetitions. Allowing another $\bar{n}_\ell$ positions for the near window gives the threshold κ_ℓ at the next layer.

The following lemma is the main obstruction: after fixing an earlier prefix, the length of a long run at the end is forgotten.

Lemma 3.3 (Terminal-run saturation)**.** *Fix a horizon N and $T_N \in \mathcal{T}_{\mathrm{ECB}}$ with depth $D = D(N)$. Let*

$$\kappa_D := \sum_{\ell=1}^{D} \left(\bar{n}_\ell + 2^{p(N)}\right), \qquad \bar{n}_\ell := \max_h n_h^{(\ell)}.$$

*For every fixed prefix $x \in \Gamma^*$, every $\sigma \in \mathcal{V}$, and every $k, k' \geq \kappa_D$ for which the contexts (x, σ^k) and $(x, \sigma^{k'})$ are in the model domain, the next-token logits on these two contexts are **identical**.*

It follows that, after a long enough terminal run of σ's, the next-token map may still depend on the earlier prefix x, but it no longer depends on the length of that run.

Theorem 3.4 ($\mathcal{T}_{\mathrm{ECB}}$ lower bound: correct copy implies a σ-loop)**.** *Assume the deterministic left-to-right finite-precision execution model of Section 3.1. Let $\{T_N\} \in \mathcal{T}_{\mathrm{ECB}}$ have depth $D(N)$ and precision $p(N)$ at design horizon N. Let $\kappa_D(N)$ be the saturation threshold from Lemma 3.3 for T_N. If $\kappa_D(N) = o(N)$, then for all sufficiently large N, every $L \in [N/3, N/2]$, and every fixed $\sigma \in \mathcal{V}$, T_N fails on the all-identical prompt*

$$\langle \textsc{Bos} \rangle, \sigma^L, \langle \textsc{Eoi} \rangle, \langle \textsc{Copy} \rangle.$$

More precisely, it either:

1. *Makes a mistake before completing the L-token copy, or,*
2. *if it outputs $y_1 = \cdots = y_L = \sigma$, then $y_r = \sigma$ for every $r > L$ for which the context remains in the model domain of T_N.*

Proof idea. For the all-identical copy task, take

$$x_L := \langle \textsc{Bos} \rangle, \sigma^L, \langle \textsc{Eoi} \rangle, \langle \textsc{Copy} \rangle.$$

The interval $L \in [N/3, N/2]$ is only a convenient linear-length range: for sufficiently large N, it ensures $L - 1 \geq \kappa_D(N)$ while keeping the prompt and generated prefix within the model domain. If the L-th copied token is correctly generated from (x_L, σ^{L-1}), then the termination query (x_L, σ^L) has the same logits and hence also outputs σ, even though the correct token is $\langle \textsc{Eos} \rangle$. Repeating the same argument gives the loop.

For the recursive ALiBi case below, the eventual-constancy argument assumes that the bias recurrence does not overflow, as in Lemma D.8. This isolates stagnation from rounding rather than failure from exponent overflow.

Corollary 3.5. *Consider the variants of ALiBi (Press et al., 2022) and Hard-ALiBi (Jelassi et al., 2024) as defined in Definition D.7. For depth-$D(N)$, arbitrary-width, deterministic $p(N)$-bit models satisfying $D(N)2^{p(N)} = o(N)$, Theorem 3.4 applies to:*

- ***NoPE:*** *no positional logit bias.*
- ***Recursive implementation of ALiBi:*** *the rounded slope recurrence $b_h(d+1) = \mathsf{fl}(b_h(d) + m_h)$.*
- ***Bounded-window Hard-ALiBi:*** *hard-masked heads whose attention windows are at most $2^{p(N)}$, with any remaining heads treated as NoPE heads.*

In particular, this applies even when $D = N^{0.01}$ and $p \leq 0.9 \log_2 N$.

Scope. Under exact accumulation, the attention mass assigned to a repeated run relative to the fixed prefix can continue to carry length information. The lower bound identifies how repeated rounding in the execution model of Section 3.1 makes this information stop changing. Its ECB assumption restricts positional access to eventually-constant additive biases, leaving absolute embeddings, non-eventually-constant ALiBi, QK rotations, and value-side PEs outside the class. Whether an analogous saturation statement holds under balanced-tree or pairwise summation remains open.

3.3 Upper Bound via VO Position Hashing

The positive construction replaces "content hashing" in Jelassi et al. (2024) by position hashing. Layer 1 selects a delimiter rather than a repeated source token and writes the query's distance from that delimiter into the value stream. Layer 2 uses this representation to select the next source occurrence, without matching token content.

Anchors and offsets. We call $\langle \textsc{Bos} \rangle$ and $\langle \textsc{Copy} \rangle$ the *anchors*. The *active anchor* at position t is

$$a(t) := \begin{cases} \text{the unique } \langle \textsc{Copy} \rangle \text{ position in } [1, t], & \text{if } \langle \textsc{Copy} \rangle \text{ has appeared by time } t, \\ \text{the } \langle \textsc{Bos} \rangle \text{ position}, & \text{otherwise.} \end{cases}$$

Thus $a(t)$ is the nearest anchor at or before t, and the *active-anchor offset* is $\delta(t) := t - a(t)$. Source token s_k has offset k, and the source-side $\langle \textsc{Eoi} \rangle$ has offset $L + 1$. At a copy-region query of offset k, the desired source-side offset is $k + 1$. All visible copy-region offsets are at most k, so this target occurs only on the source side: it holds s_{k+1} if $k < L$ and $\langle \textsc{Eoi} \rangle$ if $k = L$. The final readout maps vocabulary tokens to themselves and the retrieved $\langle \textsc{Eoi} \rangle$ to $\langle \textsc{Eos} \rangle$.

Writing offsets into values. VO-rotation encodes the distance to the selected anchor modulo a head-specific period. For a period q, let R_q be the planar rotation by $2\pi/q$. In the two-dimensional case, one-hot attention to position j with rotary value $\mathbf{e}_1$ returns $R_q^{t-j}\mathbf{e}_1$, whose coordinates are $(\cos(2\pi(t-j)/q), \sin(2\pi(t-j)/q))$, up to rounding. Thus the returned vector records the selected offset modulo q. On an even-dimensional rotary value subspace, let $\mathsf{R}_{q,m}$ denote block-diagonal copies of R_q; powers are interpreted modulo q.

Definition 3.6 (VO-rotated attention head)**.** *A VO head is an ordinary NoPE causal attention head, except that a chosen value subspace is pre-rotated by the key position and post-rotated by the query position. For* $\mathbf{v}_j^{(h)} = (\mathbf{v}_j^{(h),\mathrm{rot}}, \mathbf{v}_j^{(h),\mathrm{rest}})$ *with* $\mathbf{v}_j^{(h),\mathrm{rot}} \in \mathbb{R}^{m_h}$*, its head-local pre-output before rounding satisfies*

$$\widetilde{\mathbf{o}}_t^{(h),\mathrm{rot}} = \mathsf{R}_{q_h,m_h}^{t} \sum_{j\le t} \alpha_{t,j}^{(h)} \mathsf{R}_{q_h,m_h}^{-j} \mathbf{v}_j^{(h),\mathrm{rot}}, \qquad \widetilde{\mathbf{o}}_t^{(h),\mathrm{rest}} = \sum_{j\le t} \alpha_{t,j}^{(h)} \mathbf{v}_j^{(h),\mathrm{rest}}.$$

In exact arithmetic, one-hot attention on j *gives rotary block* $\mathsf{R}_{q_h,m_h}^{t-j}\mathbf{v}_j^{(h),\mathrm{rot}}$*, so the selected value carries the relative offset* $t-j \bmod q_h$*. Lemma D.14 bounds the error from the two rounded rotations in the construction.*

Although the definition allows general m_h, our construction uses only one two-dimensional block per head, by setting $m_h = 2$ and $\mathbf{v}_j^{(h),\mathrm{rot}} = \mathbf{e}_1$. Several prime-period heads therefore form a CRT fingerprint of the selected offset.

Definition 3.7 (VO positional transformer class $\mathcal{T}_{\mathrm{VO}}$)**.** *We write* $\mathcal{T}_{\mathrm{VO}}$ *for transformers equipped with such VO heads and exact residues modulo the chosen periods to index rounded rotation tables; Appendix D.3 gives the formal instantiation.*

Theorem 3.8 ($\mathcal{T}_{\mathrm{VO}}$ upper bound with CRT-chosen periods)**.** *For every fixed finite vocabulary* $\mathcal{V}$ *and all sufficiently large design horizons* N*, after choosing token embeddings and a fixed linear readout for* $\mathcal{V} \cup \{\langle \textsc{Bos}\rangle, \langle \textsc{EoI}\rangle, \langle \textsc{Copy}\rangle, \langle \textsc{EoS}\rangle\}$*, there exists a two-layer model family* $\{T_N\}$ *in* $\mathcal{T}_{\mathrm{VO}}$ *such that each* T_N *correctly copies every string of every length* $L \le N$*:*

$$\langle \textsc{Bos}\rangle, s_1, \ldots, s_L, \langle \textsc{EoI}\rangle, \langle \textsc{Copy}\rangle \quad \mapsto \quad s_1, \ldots, s_L, \langle \textsc{EoS}\rangle.$$

The model T_N *uses:*

- $H = \Theta(\log N/\log\log N)$ *first-layer prime-period heads, each using* $m_h = 2$*;*
- *floating-point precision* $p = O(\log\log N)$*;*
- *dimension* $d = O(|\mathcal{V}| + \log N/\log\log N)$*;*
- *the exact residue primitive in Definition 3.7.*

Proof sketch. Layer 1 attends one-hot to the active anchor: $\langle \textsc{Bos}\rangle$ before $\langle \textsc{Copy}\rangle$ and $\langle \textsc{Copy}\rangle$ afterward. VO writes $\delta(t) \bmod q_h$ into head h. Since $\prod_h q_h > N+1$, CRT identifies every offset in the design horizon. At a copy-region query of offset k, Layer 2 rotates each query fingerprint component by one step, while the keys retain the candidate positions' fingerprints. Before scaling and rounding, their centered inner products sum to

$$\sum_{h=1}^{H} \left[\cos\left(\frac{2\pi(\delta(j) - \delta(t) - 1)}{q_h} \right) - 1 \right].$$

The unique visible position at offset $k+1$ matches every residue and has score zero; every other position mismatches in at least one component and has a negative score. The precision bound preserves this gap, and scaling makes the non-winning softmax weights underflow to zero. If $k = L$, this position is the source-side $\langle \textsc{EoI}\rangle$, and the final readout outputs $\langle \textsc{EoS}\rangle$. Appendix D gives the finite-precision gap and resource accounting.

The first $H = \Theta(\log N/\log\log N)$ primes have product $\Pi := \prod_h q_h > N+1$, while the largest period is only $\Theta(\log N)$. Each head therefore separates $O(\log N)$ phases; the gap analysis shows that these distinctions survive at $O(\log\log N)$-bit precision. Exact residue access is an additional positional requirement, implemented with $O(\log\log N)$-bit indices and rounded phase coordinates within the construction's precision budget; see Section D.4.4.

4 Experiments

We examine three questions about learning to copy: how standard positional baselines respond to repetition under a shared training distribution, whether a shallow model can learn to exploit VO-rotation, and whether its long-distance behavior is consistent with the supplied prime periods. We train VO with seeds 42, 43, and 44 using the same configuration, and report seed 42 for each baseline. Architectures and training budgets differ across model families. Section B records the configurations and evaluation details.

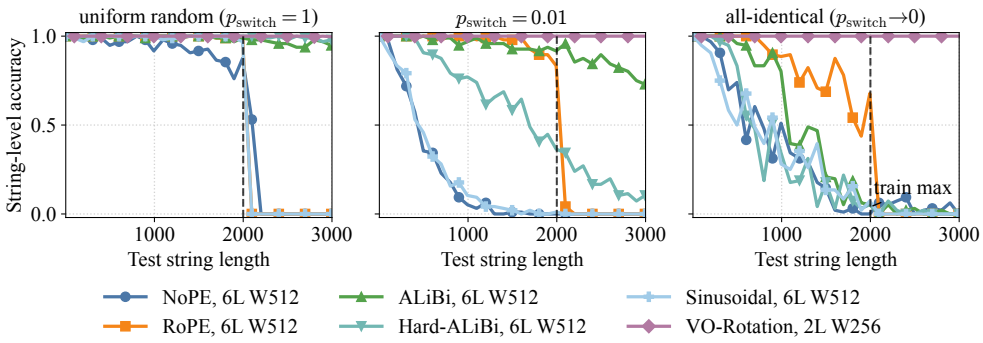


Figure 3: Exact-copy accuracy by source length and distribution. VO shows the three-seed mean and pointwise ± 1 sample standard deviation; baselines use seed 42. The dashed line marks the training limit, $n = 2000$.

4.1 Setup

Each example uses the copying task of Section 3.1 over the 26 lowercase English letters; we write n for the source length, denoted L there. The loss is applied only to the generated answer, including the end marker. Our metric is string-level exact-copy accuracy, requiring the entire answer, including termination, to match. During training, the source length is drawn as $n \sim \text{Unif}\{1, \ldots, 2000\}$, and each string is drawn from a three-component mixture: an i.i.d. string with probability 0.1, an all-identical string with probability 0.1, and, with probability 0.8, a string of same-character blocks, whose repetition level is randomized by drawing $u \sim \text{Unif}[0, 300]$ and setting $p_{\text{switch}} = 1/(u + 1)$. Section B gives the block sampler and literal delimiter encoding.

Evaluation uses i.i.d. uniform strings ($p_{\text{switch}} = 1$), repeated strings ($p_{\text{switch}} \approx 0.01$), and all-identical strings ($p_{\text{switch}} \to 0$).

4.2 Standard Non-Value-Side Positional Baselines

We first compare NoPE, RoPE, ALiBi, Hard-ALiBi, and Sinusoidal (Vaswani et al., 2017) using the same 6-layer, width-512, 8-head `bf16` backbone. The ALiBi baseline uses the directly computed MPT-style bias; because the direct bias is not eventually constant, this baseline lies outside the lower-bound class $\mathcal{T}_{\text{ECB}}$ and serves as an empirical reference point. Figure 3 shows how these baselines respond to the three repetition conditions. The contrast is sharpest near the training limit: at $n = 1993$, every non-value-side baseline drops below 0.7 on the all-identical diagnostic (Table 1), while the two-layer VO model reaches 1.000 in all three seeds at the nearest evaluated length, $n = 1992$. The comparison concerns the learned models under this protocol, not impossibility for their architectural classes.

4.3 VO-Rotation

We next test whether training can exploit the value-side positional encoding. The VO-Rotation models use a 2-layer, width-256, 4-head backbone with simulated `fp8` rounding of linear-layer inputs and weights, value/output rotation, and no score-side positional term. This simulation is not the all-operations p-bit execution model of the theorem. Each head carries a single prime period $(5, 7, 11, 13)$, matching the construction's one-period-per-head layout. The joint residue vector repeats every 5005 offsets. The seed-42 training loss stays on a long plateau before dropping sharply (Figure 5).

The models share the construction's periodic value-side encoding, but their weights are learned rather than set to the explicit construction in Theorem 3.8. Their performance shows that training can find successful copying solutions in this shallow architecture under the reported precision simulation.

Table 1: String-level accuracy near the $n = 2000$ training limit. Baselines use seed 42 at $n = 1993$; VO attains the reported values in all three seeds at $n = 1992$.

Model	Layers	Width	$p_{\text{switch}} = 1$	$p_{\text{switch}} = 0.01$	$p_{\text{switch}} \to 0$
NoPE	6	512	0.875	0.000	0.000
RoPE	6	512	0.990	0.833	0.677
ALiBi (MPT-style)	6	512	0.990	0.917	0.052
Hard-ALiBi	6	512	1.000	0.365	0.052
Sinusoidal	6	512	0.969	0.010	0.073
VO-Rotation	2	256	1.000	1.000	1.000

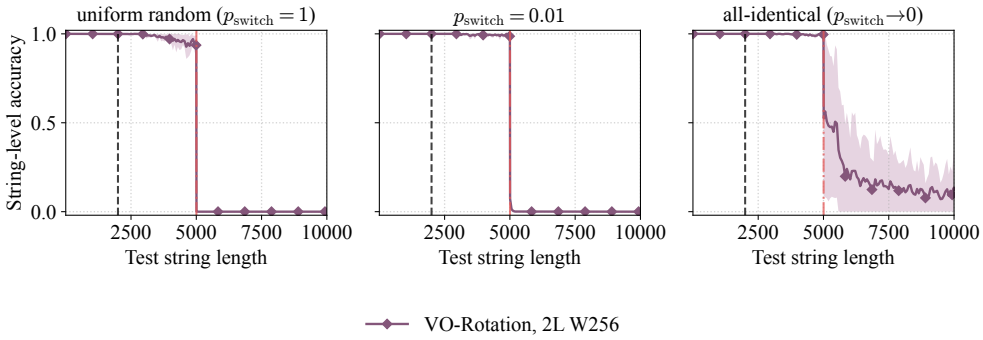


Figure 4: VO-Rotation length generalization: mean ± 1 sample standard deviation across three seeds after 9-point smoothing and sparse plotting (band clipped to $[0, 1]$). Dashed black: training limit 2000; dash-dotted red: period product 5005. Raw curves and display details: Figure 6.

Generalization and the period product. Every seed achieves perfect accuracy at all evaluated lengths through 2000 in all three conditions (Figure 4). Over extrapolation lengths 2008–5000, mean accuracy across lengths and seeds is 0.980 for uniform-random, 0.996 for moderately repeated, and 0.998 for all-identical strings. Uniform-random and moderately repeated accuracy drops sharply just beyond the period product $5 \cdot 7 \cdot 11 \cdot 13 = 5005$ in every seed. Offsets separated by 5005 share residues, so residue-only retrieval cannot distinguish them. This transition supports learned use of modular positions, but does not identify the computation with the exact construction. All-identical accuracy beyond 5005 varies across seeds (Section B.1).

5 Conclusion

Repetition makes content-based occurrence matching ambiguous and motivates representing positions relative to delimiters. VO-rotation writes anchor-relative offsets into the value stream. With exact modular-residue access and CRT periods, a two-layer model copies arbitrary strings throughout its design horizon, using the same retrieval rule for copying and termination at $O(\log \log N)$-bit precision. The $\mathcal{T}_{\text{ECB}}$ lower bound identifies a width-independent finite-precision obstruction: under the stated saturation condition, a model cannot both finish a sufficiently long identical copy and terminate correctly. The synthetic runs show that training can exploit a related value-side encoding, and the observed transition near the period product supports the modular-position interpretation without identifying the learned computation with the proof.

The broader design question is how positional information should reach later layers under repetition. Extending the lower bound to other accumulation rules or comparing score-side and value-side access to the same residues remains open. The frontier-model diagnostic motivates that question, but lies outside the ECB theorem and does not identify the cause of those failures. Hybrid VO/RoPE training and the effect of value-side rotations on tasks that rely on value semantics remain to be tested.

6 AI DISCLOSURE

In this work, we used GPT and Claude to assist with key proof steps, writing and organizing proofs, implementing code, and checking and interpreting experimental results. The overarching research ideas originated with the authors. Additionally, we used these tools to prepare figures and improve the clarity of the manuscript. The authors checked the proofs, verified references, ran the code, and reviewed the final text. We take responsibility for the final manuscript, including all AI-assisted text, claims, code, and figures.

REFERENCES

Thomas Chen, Tengyu Ma, and Zhiyuan Li. Non-asymptotic length generalization. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*, pp. 8621–8682. PMLR, 2025. URL https://proceedings.mlr.press/v267/chen25ar.html.

Ying Fan, Yilun Du, Kannan Ramchandran, and Kangwook Lee. Looped transformers for length generalization. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=2edigk8yoU.

Alejandro García-Castellanos, Maurice Weiler, and Erik J. Bekkers. RoVE: Rotary value embeddings attention for relative position-dependent value pathways. *arXiv preprint arXiv:2606.11275*, 2026. URL https://arxiv.org/abs/2606.11275.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. doi: 10.18653/V1/P16-1154. URL https://doi.org/10.18653/v1/p16-1154.

Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulić. ConveRT: Efficient and accurate conversational representations from transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 2161–2174. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.findings-emnlp.196. URL https://aclanthology.org/2020.findings-emnlp.196/.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, Xinrong Zhang, Zheng Leng Thai, Kaihuo Zhang, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. MiniCPM: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024. URL https://arxiv.org/abs/2404.06395.

Xinting Huang, Andy Yang, Satwik Bhattamishra, Yash Sarrof, Andreas Krebs, Hattie Zhou, Preetum Nakkiran, and Michael Hahn. A formal framework for understanding length generalization in transformers. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=U49N5V51rU.

Samy Jelassi, David Brandfonbrener, Sham M. Kakade, and Eran Malach. Repeat after me: Transformers are better than state space models at copying. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 21502–21521. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/jelassi24a.html.

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. The impact of positional encoding on length generalization in transformers. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural*

Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/4e85362c02172c0c6567ce593122d31c-Abstract-Conference.html.

Oliver Kraus, Yash Sarrof, Yuekun Yao, Alexander Koller, and Michael Hahn. Barriers to universal reasoning with transformers (and how to overcome them). *arXiv preprint arXiv:2604.25800*, 2026. URL https://arxiv.org/abs/2604.25800.

Jiaoda Li and Ryan Cotterell. Characterizing the expressivity of fixed-precision transformer language models. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2505.23623.

Jiaoda Li and Ryan Cotterell. Characterizing the expressivity of local attention in transformers. *arXiv preprint arXiv:2605.00768*, 2026. URL https://arxiv.org/abs/2605.00768.

Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. Chain of thought empowers transformers to solve inherently serial problems. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=3EWTEy9MTM.

William Merrill and Ashish Sabharwal. The parallelism tradeoff: Limitations of log-precision transformers. *Trans. Assoc. Comput. Linguistics*, 11:531–545, 2023. doi: 10.1162/TACL_A_00562. URL https://doi.org/10.1162/tacl_a_00562.

Takeru Miyato, Bernhard Jaeger, Max Welling, and Andreas Geiger. GTA: A geometry-aware attention mechanism for multi-view transformers. In *International Conference on Learning Representations*, 2024. URL https://takerum.github.io/gta/.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *CoRR*, abs/2209.11895, 2022. doi: 10.48550/ARXIV.2209.11895. URL https://doi.org/10.48550/arXiv.2209.11895.

Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=R8sQPpGCv0.

Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pp. 1073–1083. Association for Computational Linguistics, 2017. doi: 10.18653/V1/P17-1099. URL https://doi.org/10.18653/v1/P17-1099.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In Marilyn A. Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pp. 464–468. Association for Computational Linguistics, 2018. doi: 10.18653/V1/N18-2074. URL https://doi.org/10.18653/v1/n18-2074.

Lena Strobl, William Merrill, Gail Weiss, David Chiang, and Dana Angluin. What formal languages can transformers express? A survey. *Trans. Assoc. Comput. Linguistics*, 12:543–561, 2024. doi: 10.1162/TACL_A_00663. URL https://doi.org/10.1162/tacl_a_00663.

Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/J.NEUCOM.2023.127063. URL https://doi.org/10.1016/j.neucom.2023.127063.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 2692–2700, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/29921001f2f04bd3baee84a12e98098f-Abstract.html.

Andy Yang, Michaël Cadilhac, and David Chiang. Knee-deep in C-RASP: A transformer depth hierarchy. *CoRR*, abs/2506.16055, 2025. doi: 10.48550/ARXIV.2506.16055. URL https://doi.org/10.48550/arXiv.2506.16055.

Andy Yang, Pascal Bergsträßer, Georg Zetzsche, David Chiang, and Anthony Widjaja Lin. Length generalization bounds for transformers. *CoRR*, abs/2603.02238, 2026. doi: 10.48550/ARXIV.2603.02238. URL https://doi.org/10.48550/arXiv.2603.02238.

Hattie Zhou, Arwen Bradley, Etai Littwin, Noam Razin, Omid Saremi, Joshua M. Susskind, Samy Bengio, and Preetum Nakkiran. What algorithms can transformers learn? A study in length generalization. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=AssIuHnmHX.

A Details of the LLM Copying Experiment

A.1 Prompt and Experimental Setup

Each model is instructed via the following system prompt:

> *"You are a precise string-copying assistant. You will see a string enclosed between '$' and '.', followed by ':'. Output ONLY the enclosed string, character by character, with absolutely nothing else – no spaces before or after, no explanation, no newline."*

The user prompt has the format

$$\texttt{\$}\quad \langle \text{string} \rangle \quad \texttt{.}\quad \texttt{:}$$

where `.` is the prompt end delimiter and `:` is the copy cue. The model is tasked to output exactly the enclosed string, not the delimiters.

Output extraction and scoring. String-level accuracy compares the stored extracted prediction with the target, not the raw response with the full prompt instructions. The archived extraction routine removes `<think>` blocks, splits the remaining response into nonempty lines, and removes answer prefixes, delimiter symbols, quotes, backticks, and spaces. Among cleaned lines containing only lowercase letters, it selects the one whose length is closest to the target length; if none exists, it uses the cleaned last line. Thus target length participates in extraction, and the metric does not assess strict compliance with the no-extra-text instruction. The figures use the recorded exact-match scores, including recorded generation-cap and API failures, rather than rescoring responses with a new extraction rule.

Model variants. Table 2 identifies the model IDs and reasoning settings for each evaluated variant. We report these variants separately, without treating their comparison as an isolated test of reasoning.

Table 2: Frontier-model variants plotted in Figure 1. Each row uses 50 trials for each length and condition, a maximum generation budget of 20,000 output tokens, and the extraction policy described above. Provider length-cap finishes such as `MAX_TOKENS` or `finish_reason=length` are retained in the CSVs and counted as exact-copy failures.

CSV variant	Actual model ID	Reasoning/thinking
`397b`	`qwen3.5:397b-cloud`	thinking disabled
`deepseek-v4-pro-no-thinking`	`deepseek-v4-pro`	thinking disabled
`gemini-3.1-pro-low-thinking`	`gemini-3.1-pro-preview`	low thinking
`gemini-3.1-pro-medium-thinking`	`gemini-3.1-pro-preview`	medium thinking
`gpt-5.5`	`gpt-5.5`	none
`gpt-5.5-medium-reasoning`	`gpt-5.5`	medium reasoning

Lengths and repetition conditions. We evaluate source strings of 30, 100, 300, 1000, and 3000 characters under three conditions: i.i.d. uniform lowercase strings, geometric same-character blocks, and all-identical strings. For the block condition, each character is sampled uniformly and independently and its block length is geometric, truncated at the requested source length. The recorded condition `geometric_L=100.0` uses block parameter 100, giving $p_{\text{switch}} = 1/(100+1) \approx 0.01$; Figure 1 abbreviates this as $p_{\text{switch}} = 0.01$. Adjacent sampled blocks may share a character. Figure 1 reports the stored string-level accuracy averaged over trials.

A.2 Failure Modes of Frontier LLMs

Observed repeated-content failures include extending a block beyond the target, returning too few characters, and recording no copied characters. These are output patterns, not a classification of the underlying causes; in particular, an empty prediction alone does not distinguish a model stopping decision from an API or generation-budget failure.

Table 3 provides representative examples generated by Qwen3.5 on all-identical sequences with length $L = 300$.

Failure Type	**Target String (Expected Length)**	**Model Output (Actual Length)**
Extended repetition	bbbbbbbbbbbbbbbbbbbbbb... (Length: 300, all-identical)	bbbbbbbbbbbbbbbbbbbbbb... (Model continued until the generation limit at **79,984** b's)
Premature truncation	hhhhhhhhhhhhhhhhhhhhhh... (Length: 300, all-identical)	hhhhhhhhhhhhhhhhhhhhhh... (Model halts early at **267** characters)
No output	zzzzzzzzzzzzzzzzzzzzzz... (Length: 300, all-identical)	[Empty String] (No copied character is recorded)

Table 3: Representative string-copying failures observed with Qwen3.5 on repeated-content sequences. The outputs illustrate failures to reproduce the target length, without identifying their internal cause.

A.3 Relation to the Theoretical Task

The diagnostic establishes copying failures under this prompting and evaluation setup, not each model's maximum copying ability or the cause of its errors. The evaluated models have positional mechanisms outside the lower-bound class, so their failures motivate the expressivity question rather than test the ECB theorem.

The experiment also differs from the theory in token granularity: prompts repeat characters, whereas the theorem concerns repeated source tokens. Subword tokenizers can segment identical-character runs into repeated block tokens, with length-dependent boundary tokens. For example, under `o200k_base`, a run of 300 `b`'s becomes 75 copies of the four-character token `bbbb`. This example

shows how character repetition can induce token repetition; it does not identify the tokenization of every evaluated model or establish token-level equivalence for the full diagnostic.

B Additional Synthetic Experiment Details

Task encoding and string generation. The synthetic implementation uses the literal tokens `$`, `;`, `:`, and `.` for ⟨BOS⟩, ⟨EOI⟩, ⟨COPY⟩, and ⟨EOS⟩, respectively:

$$\texttt{\$}, x_1, \ldots, x_n, \texttt{;}, \texttt{:} \mapsto x_1, \ldots, x_n, \texttt{.}$$

The loss and exact-copy metric cover the generated answer after `:`, including the final period token. Training draws $n \sim \mathrm{Unif}\{1, \ldots, 2000\}$ and uses i.i.d., all-identical, and block-generated strings with probabilities 0.1, 0.1, and 0.8, respectively. For the block component, we draw $u \sim \mathrm{Unif}[0, 300]$ and set $p_{\mathrm{switch}} = 1/(u+1)$. For each block, we independently draw a character uniformly from the alphabet and a length from $\mathrm{Geom}(p_{\mathrm{switch}})$ on the positive integers, truncating the last block to reach source length n. Adjacent blocks share the same character with probability $1/26$, so a maximal same-character run may span several sampled blocks. The repeated evaluation condition uses block parameter $u = 100$, giving $p_{\mathrm{switch}} = 1/(100+1) \approx 0.01$; figures and tables use the short label $p_{\mathrm{switch}} = 0.01$.

Training configurations. Table 4 identifies the runs and checkpoints used in Section 4. VO uses training seeds 42, 43, and 44 with a shared configuration; each reported baseline uses seed 42. Each seed initializes Python, NumPy, and PyTorch.

All rows use the reported training configurations in Table 4: effective batch size 32 (microbatch 4 with gradient accumulation 8 for the `bf16` baselines, microbatch 8 with gradient accumulation 4 for VO-Rotation), GELU MLPs, layer normalization, warmup-stable-decay (WSD) learning-rate scheduling (Hu et al., 2024) with 2000 warmup optimizer updates and cooldown fraction 0.3, and EMA tracking with decay 0.999. The standard positional-baseline rows use 8 attention heads of dimension 64. The ALiBi baseline uses the directly computed MPT-style bias[2] with `alibi_bias_max` $= 8$; for 8 heads, the slopes are $m_h = 2^{-h}$, $h = 1, \ldots, 8$, so up to row-wise softmax constants the bias for query position i and key position $j \le i$ is $-m_h(i-j)$. Hard-ALiBi uses `num_masked_heads` $= 4$: for masked head $h = 1, \ldots, 4$, only keys with backward offset $i - j \in \{0, \ldots, h-1\}$ remain visible, while heads 5–8 keep the ordinary causal mask. The `fp8` models use a simulated FP8 forward pass implemented by rounding linear-layer inputs and weights. Each VO model uses one modified VO layer with four prime-period heads, one head per prime; the periods are the first four primes starting at 5, namely $5, 7, 11, 13$. This follows the one-period-per-head structure of Theorem 3.8, but the rotary value subspace is wider than the $m_h = 2$ the theorem uses (16 of the 64 coordinates per head), and the vector written into it is learned rather than fixed to $\mathbf{e}_1$. All rotary blocks within a head share the single frequency $2\pi/q_h$, so the extra width provides learned value components at that frequency rather than additional periods. These architectural correspondences do not identify the learned weights with the theorem's construction. The configured training-step column reports the script argument passed as `-steps`; optimization plots use the logged optimizer-update index instead, because WSD may enter cooldown and terminate before the configured horizon. All three VO runs trigger cooldown when the EMA training loss (decay 0.98) falls below 10^{-4}.

Evaluation and recorded metrics. The W&B metric names are `eval_string_accuracy_*` and `eval_char_accuracy_*`; the suffix identifies latest-checkpoint or EMA evaluation. Section 4 reports string-level accuracy from latest-checkpoint metrics. Token-level character accuracy is evaluated on the same generated answer region but is not used as the main plotted metric. For each evaluated length, the exports average over 32 repeats, 3 evaluation batches, and batch size 1, for 96 examples per point and training seed. The baseline evaluations use source lengths $1, 9, 17, \ldots, 2993$; the VO-only stress evaluation uses $8, 16, \ldots, 10{,}000$. Figure 3 shows a sparse length grid, matching VO points to baseline lengths within four tokens. VO curves and shaded bands show the mean and pointwise sample standard deviation across the three training seeds, with bands clipped to $[0, 1]$. Figure 4 smooths each seed before aggregation and uses a sparse grid (Figure 6); Figure 3 uses unsmoothed values. These bands describe cross-seed variation, not confidence intervals. Figure 6

[2] MPT-7B attention implementation.

Table 4: Run provenance for the synthetic experiments. "Final update" is the last logged optimizer-update index in the local training-history export used for the corresponding checkpoint. For seeds 43/44, starred values instead give completed optimizer updates from the training-completion records.

Model	W&B project/run	Layers	Width	Heads	Precision	LR / WD	Cfg. steps	Final update	Eval ckpt.
NoPE	`baseline_nope/uku6u8vv`	6	512	8	`bf16`	10^{-4} / 0	2.56M	102,800	latest
RoPE	`baseline_rope/zm4kpn33`	6	512	8	`bf16`	10^{-4} / 0	2.56M	144,800	latest
ALiBi (MPT-style)	`baseline_alibi/wyp6xfr2`	6	512	8	`bf16`	10^{-4} / 0	2.56M	81,200	latest
Hard-ALiBi	`baseline_hard_alibi/pugs7dsx`	6	512	8	`bf16`	10^{-4} / 0	2.56M	110,200	latest
Sinusoidal	`baseline_sinusoidal/0t1j6vyp`	6	512	8	`bf16`	10^{-4} / 0	2.56M	113,800	latest
VO (seed 42)	`VO_no_scalar_/6ezr3zh5`	2	256	4	`fp8`	10^{-4} / 0	1.60M	158,200	latest
VO (seed 43)	`VO_no_scalar_/5umy26gh`	2	256	4	`fp8`	10^{-4} / 0	1.60M	150,020*	latest
VO (seed 44)	`VO_no_scalar_/6l6gelzl`	2	256	4	`fp8`	10^{-4} / 0	1.60M	148,200*	latest

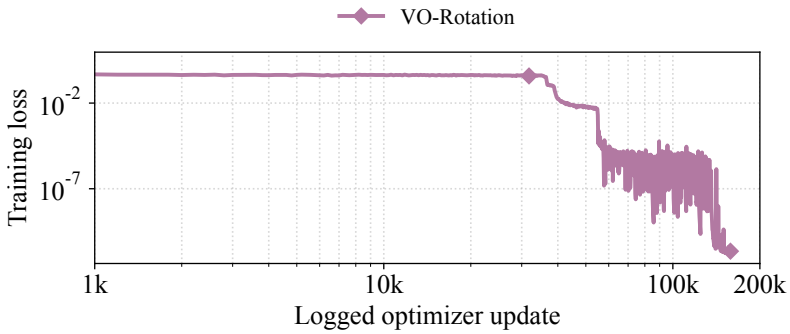


Figure 5: Training loss of VO-Rotation with seed 42, plotted from the local W&B history export on a logarithmic scale. The loss stays on a plateau for roughly the first 40k optimizer updates and then drops sharply.

shows the individual curves. The local W&B history exports also provide the seed-42 `train_loss` trajectories plotted in Figure 5.

Table 1 gives the near-training-limit comparison summarized in Section 4.2.

Compute and provenance. Table 4 lists the training project and run identifiers. Seed-42 evaluations are in `synthetic_expt_data/wandb_results/`; the merged seed-43/44 CSV, including evaluation run IDs and source-log paths, is in `synthetic_expt_data/VO_no_scalar_10k_seed43_seed44.csv`. The merged export retains the complete grid across interrupted and resumed evaluations. The synthetic training jobs were launched on a Slurm-managed GPU partition with one GPU and four CPU cores per job. The standard positional-encoding baselines used NVIDIA RTX A6000 jobs; all three VO-Rotation training runs used NVIDIA RTX A4000 jobs.

B.1 VO-Rotation Length Generalization up to 10,000 and the Period Product

Each final VO checkpoint is evaluated at $n = 8, 16, \ldots, 10{,}000$, with 96 examples per length and condition. All three seeds score 1.000 at every evaluated $n \leq 2000$ in every condition. Over the 375 extrapolation lengths 2008–5000, per-seed length averages have across-seed mean and sample standard deviation 0.9796 ± 0.0189 (uniform random), 0.9963 ± 0.0058 ($p_{\text{switch}} = 0.01$), and 0.9978 ± 0.0020 (all-identical). Table 5 gives individual means and minima.

At 5008, the first evaluated length past the period product, mean accuracy falls to 0.0035 (uniform random) and 0.2500 (moderately repeated). Over the 625 lengths 5008–10,000, their means are below 0.00001 and 0.001, respectively. All-identical behavior varies substantially (Figure 6): seeds 42, 43, and 44 score 0.7292, 0.9063, and 0.0313 at 5008, with respective post-product means 0.4196, 0.1242, and 0.0162.

Relation to the period representation. The residue vector repeats every 5005 offsets, so residue-only retrieval cannot distinguish positions separated by that product. The decline at 5008 agrees with

Table 5: VO exact-copy accuracy by training seed and distribution. Mean and minimum are over the 375 evaluated lengths 2008–5000; the final column gives accuracy at 5000.

Seed	Distribution	Mean	Minimum	$n = 5000$
42	Uniform random	0.9834	0.4375	0.9688
43	Uniform random	0.9591	0.4688	0.8854
44	Uniform random	0.9964	0.9583	1.0000
42	$p_{\text{switch}} = 0.01$	0.9993	0.9167	1.0000
43	$p_{\text{switch}} = 0.01$	0.9896	0.7188	0.9896
44	$p_{\text{switch}} = 0.01$	0.99997	0.9896	1.0000
42	All-identical	0.9962	0.9063	1.0000
43	All-identical	0.9973	0.8542	1.0000
44	All-identical	1.0000	1.0000	1.0000

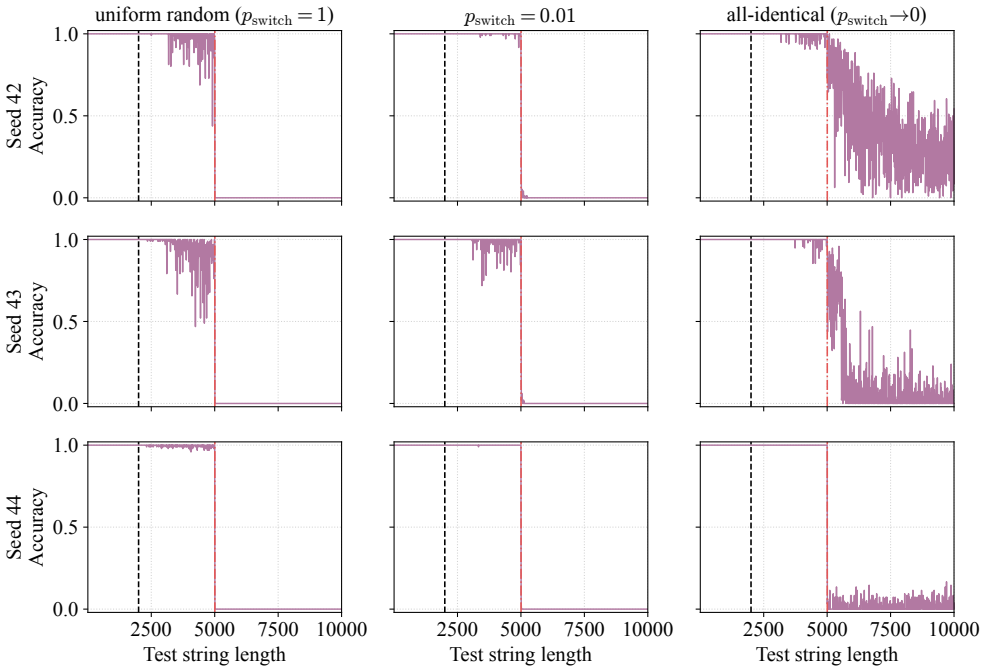


Figure 6: Unsmoothed VO-Rotation curves on the complete evaluation grid, with seeds in rows and distributions in columns. Dashed black and dash-dotted red lines mark 2000 and 5005. For Figure 4, we first apply a centered 9-point moving average to each seed, truncating windows at these two boundaries and the grid endpoints, then compute the across-seed mean and sample standard deviation. The mean and band use every eighth grid point (64-token spacing), plus 1992, 2000, 2008, 4992, 5000, 5008, 5016, and 10,000. All numerical summaries use unsmoothed data.

this period-dependent scale rather than the 2000 training limit. It supports learned use of the joint prime-period representation, but does not establish that the internal computation is the exact CRT construction or that no other cues contribute.

Confusing two letter positions can change the required token on a nonconstant source but not on an all-identical one; confusing a letter with the end delimiter can still change termination. This can explain surviving all-identical accuracy beyond the period product without a uniform gradual decline.

Training time and generalization. For seed 42, the reported in-training evaluations reach 100% accuracy by roughly 56k optimizer updates. Its stress evaluation uses the final checkpoint at 158k. Without intermediate stress sweeps, these observations do not determine when generalization toward the period product emerged or whether the additional training was necessary for it.

C Next-Token Copying at Fixed Context Length is in DLOGTIME-uniform AC^0

At fixed context length n, the next-token copy map is a delimiter-indexed multiplexer. This is the circuit fact used in Section 2; the input size fixes the current context length, not the source length.

Fix a finite source vocabulary $\mathcal{V}$ and let $\Gamma = \mathcal{V} \cup \{\langle \textsc{BoS}\rangle, \langle \textsc{EoI}\rangle, \langle \textsc{Copy}\rangle, \langle \textsc{EoS}\rangle\}$. Encode tokens by one-hot indicators $X_{i,\sigma}$, where $X_{i,\sigma} = 1$ means that the token at position i is σ. For a fixed context length n, consider copying prefixes reachable in the ordinary autoregressive copy task, i.e., prefixes that arise when the model has copied correctly so far:

$$\langle \textsc{BoS}\rangle,\ s_1, \ldots, s_L,\ \langle \textsc{EoI}\rangle,\ \langle \textsc{Copy}\rangle,\ s_1, \ldots, s_k, \qquad 0 \le k \le L,$$

with total length $n = L + 3 + k$. If the $\langle \textsc{Copy}\rangle$ token is at position $c = L + 3$, then $k = n - c$ and the correct next token is obtained from the source-side position $j(c) = n - c + 2$. This position is s_{k+1} when $k < L$ and is $\langle \textsc{EoI}\rangle$ when $k = L$. The output transduction τ sends vocabulary tokens to themselves and sends $\langle \textsc{EoI}\rangle$ to $\langle \textsc{EoS}\rangle$.

For each candidate delimiter position c, define $D_c = X_{c-1,\langle \textsc{EoI}\rangle} \wedge X_{c,\langle \textsc{Copy}\rangle}$. For later reference in the circuit layout, write

$$E_{c,\rho} := D_c \wedge X_{n-c+2,\rho}, \qquad \rho \in \mathcal{V} \cup \{\langle \textsc{EoI}\rangle\}.$$

For every output token $\sigma \in \Gamma$, define

$$Y_\sigma = \bigvee_{\substack{c \in I_n,\ \rho \in \mathcal{V} \cup \{\langle \textsc{EoI}\rangle\}: \\ \tau(\rho) = \sigma}} E_{c,\rho}, \qquad I_n = \left\{ c : \left\lceil \frac{n+3}{2} \right\rceil \le c \le n \right\}.$$

Equivalently, $I_n = \{c : 2 \le n - c + 2 \le c - 1\}$, the range in which the selected cell lies on the source side.

Proposition C.1 (Autoregressive copying at fixed context length is in DLOGTIME-uniform AC^0)**.** *The family of next-token maps on ordinary autoregressive copy-task prefixes, indexed by context length n, is computable by a DLOGTIME-uniform AC^0 circuit family of depth $O(1)$ and size $O(n|\Gamma|)$.*

Proof. On this reachable-prefix domain, exactly one delimiter gate D_c is active, at the true $\langle \textsc{EoI}\rangle, \langle \textsc{Copy}\rangle$ boundary. Therefore, for each output token σ, the OR defining Y_σ reduces to the source-side indicator selected by $X_{n-c+2,\rho}$ and then passed through the fixed transduction τ. At the final copying step this selected position is the source-side $\langle \textsc{EoI}\rangle$, so termination is handled by the same multiplexer and the fixed map $\tau(\langle \textsc{EoI}\rangle) = \langle \textsc{EoS}\rangle$.

Each D_c is a constant fan-in AND, each term $D_c \wedge X_{n-c+2,\rho}$ is a constant fan-in AND, and the final OR has fan-in $O(n)$. Thus the circuit has constant depth and size $O(n|\Gamma|)$. If tokens are represented by any fixed binary encoding rather than one-hot indicators, the predicates $X_{i,\sigma}$ are constant-size equality tests because $|\Gamma|$ is fixed, which changes only constant factors and constant depth. The direct-connection language is decidable in $O(\log n)$ time. Gate indices encode one of four constant-many types: input gates $X_{i,\sigma}$, delimiter gates D_c, term gates $E_{c,\rho}$, and output gates Y_σ. Given two gate indices, adjacency is determined by constant-many comparisons and the affine relation $i = n - c + 2$. Since these operations on $O(\log n)$-bit indices are decidable in $O(\log n)$ time, the family is DLOGTIME-uniform AC^0. □

Remark C.2 (Fixed-context circuits and horizon-uniform models)**.** *If the source length and generation offset are fixed as well, the map degenerates to a single wire from the corresponding source coordinate. The next-token map above instead selects among source positions using the delimiter-indexed table $c \mapsto n - c + 2$ compiled into the wiring. Theorem 3.4 concerns a different constraint: one transformer must copy throughout its design horizon under the specified ECB restrictions and finite-precision execution, with saturation threshold $o(N)$.*

D Proofs for Section 3

We use the copying task of Section 3.1 and the anchor-relative offsets of Section 3.3.

Proof dependencies.

ECB	Repeated-addition eventual constancy → constant-block invariance → terminal-run saturation → Theorem 3.4.
VO with CRT periods	Underflow threshold → Layer-1 anchor → CRT fingerprint → Layer-2 margin/logit gap → Layer-2 one-hot → Theorem 3.8.

Notation.

Symbol	Meaning	Definition or convention
2^p	upper bound on p-bit execution states	Section D.1
$\mathcal{M}_p$	underflow threshold for $\exp(-m)$	Definition D.1
$\bar{n}_\ell$	largest eventual-constancy index in layer ℓ	Definition D.5
κ_ℓ	terminal-run saturation threshold after layer ℓ	Lemma D.6
$R_{x,k}(\sigma)$	fixed prefix followed by a terminal σ-run	Section D.2
$H^{(\ell)}_{x,\sigma}(j)$	layer-ℓ state of the j-th terminal-run token	Section D.2
$\widehat{s}_k$	k-th token in the terminal σ-run	Section D.2
$a(t)$	active anchor at position t	Section 3.3
$\delta(t)$	active-anchor offset $t - a(t)$	Section 3.3
$\rho_q, \widehat{R}_q$	ideal rotary vector and rounded rotation table	Section D.3
$\Phi(t)$	computed CRT fingerprint of the active-anchor offset	Section D.3
q_h	prime period of head h	Definition D.11
Π	product of prime periods	Definition D.12
$q_{\max}$	largest prime period	Definition D.12
η	Layer-2 summation rounding budget	Equation (5)
γ	Layer-2 effective logit gap	Equation (5)
A	Layer-2 power-of-two logit scale	Equation (5)

D.1 Finite-Precision Primitives

Fix a p-bit floating-point format with 1 sign bit, p_e exponent bits, and p_m mantissa bits ($1+p_e+p_m = p$). Write $\mathsf{fl}(\cdot)$ for round-to-nearest-even. Let $\mathcal{F}_p$ be the finite set of representable floats, including signed zero. There are at most 2^p possible p-bit encodings, and we write $\varepsilon_{\mathrm{mach}} := 2^{-p_m}$. For lower-bound executions, we assume all arithmetic intermediates remain finite and representable. In particular, the recursive ALiBi case uses a bias recurrence without overflow (Lemma D.8). A repeated-addition recurrence also becomes constant if it reaches a deterministic absorbing state, but this observation alone does not specify how softmax handles overflow.

Definition D.1 (Underflow threshold)**.**

$$\mathcal{M}_p := \min\{m \in \mathcal{F}_p : m > 0 \text{ and } \mathsf{fl}(\exp(-m)) = 0\} = \Theta(2^{p_e}).$$

Thus $\mathcal{M}_p$ is itself representable, and so are $-\mathcal{M}_p$ and $-2\mathcal{M}_p$ whenever finite intermediates are assumed.

Fact D.2 (Repeated-addition eventual constancy)**.** *Let u be a p-bit float, S_0 an arbitrary p-bit float, and $S_{n+1} = \mathsf{fl}(S_n + u)$. If every S_n remains finite, then $S_{n+1} = S_n$ for all $n \geq n^*$ for some $n^* \leq 2^p$.*

Proof. For $u = 0$ the claim is immediate. For $u > 0$, round-to-nearest-even satisfies $\mathsf{fl}(x + u) \geq x$ for every finite representable x, so (S_n) is non-decreasing. A non-decreasing sequence in the finite set $\mathcal{F}_p$ becomes constant after at most 2^p distinct p-bit states. The case $u < 0$ follows by applying the $u > 0$ case to $T_n := -S_n$. □

Corollary D.3 (Componentwise eventual constancy)**.** *Let $\mathbf{u} \in \mathcal{F}_p^m$ be fixed and $(\mathbf{S}_{n+1})_c = \mathsf{fl}((\mathbf{S}_n)_c + u_c)$. Then $\mathbf{S}_n$ is eventually constant, with eventual-constancy index $n^* = \max_c n_c^* \leq 2^p$.*

Proof. Apply Fact D.2 to each coordinate and take the maximum of the coordinatewise eventual-constancy indices. □

Attention execution model. Each head at layer ℓ computes, for query position i, logits $e_{i,j}$ for $j \le i$, maximum logit $e_{\max}$, shifted weights $\hat{w}_j = \mathsf{fl}(\exp(e_{i,j} - e_{\max}))$, and a left-to-right accumulation of the softmax denominator Z and numerator $\mathbf{N}$:

$$Z_j = \mathsf{fl}(Z_{j-1} + \hat{w}_j), \qquad (\mathbf{N}_j)_c = \mathsf{fl}\big((\mathbf{N}_{j-1})_c + \mathsf{fl}(\hat{w}_j (v_j)_c)\big).$$

The head-local pre-output is $\widetilde{\mathbf{o}}_i = \mathbf{N}_i / Z_i$, and the layer update is

$$h_i^{(\ell)} := h_i^{(\ell-1)} + W_O^{(\ell)} \mathrm{Concat}\big(\widetilde{\mathbf{o}}_i^{(\ell,1)}, \dots, \widetilde{\mathbf{o}}_i^{(\ell,H)}\big) + \mathrm{MLP}^{(\ell)}(\cdots). \tag{1}$$

All operations are deterministic in p-bit arithmetic.

Lemma D.4 (Constant-block invariance)**.** *Consider a left-to-right softmax accumulation that processes a contiguous block of at least 2^p positions, each contributing the same shifted weight w and weighted value $w \cdot \mathbf{v}$. After the first 2^p positions in this block, the accumulator state is fixed. Inserting or removing additional copies of the same addend inside this constant repeated block does not change the state upon exit from the block.*

Proof. Apply Fact D.2 to the scalar denominator Z. For the numerator, apply Corollary D.3 coordinatewise to the fixed vector addend $w \cdot \mathbf{v}$. After at most 2^p identical addends, both the denominator and every numerator coordinate have reached a fixed point for this repeated addition. Any additional copies of the same addend therefore leave the accumulator unchanged. □

How these primitives are used. The 2^p state bound is used only to guarantee eventual constancy after repeated identical addends. The underflow threshold $\mathcal{M}_p$ is used only to force exact one-hot attention when all non-winners have shifted logits at most $-\mathcal{M}_p$. The machine precision $\varepsilon_{\mathrm{mach}}$ is used only to bound rounding errors in the CRT fingerprint and Layer-2 logit-gap argument.

D.2 $\mathcal{T}_{\mathrm{ECB}}$ Lower Bound

Throughout this subsection, fix the horizon N and write $p = p(N)$ for the precision at that horizon.

Terminal-run contexts. For a fixed prefix $x \in \Gamma^*$, token $\sigma \in \mathcal{V}$, and $k \ge 0$, define $R_{x,k}(\sigma) = (x, \sigma^k)$. For $k \ge 1$, let $\widehat{s}_k$ denote the k-th token in the terminal σ-run of $R_{x,k}(\sigma)$. We write

$$H_{x,\sigma}^{(\ell)}(j) := h_{\mathrm{last}}^{(\ell)}(R_{x,j}(\sigma))$$

for the layer-ℓ hidden state of the last token in $R_{x,j}(\sigma)$. By causality, this is also the state of the j-th terminal-run token inside $R_{x,k}(\sigma)$ for any $k \ge j$.

Definition D.5 (Formal $\mathcal{T}_{\mathrm{ECB}}$ model)**.** *Fix a horizon N and write D for the depth of T_N. The class $\mathcal{T}_{\mathrm{ECB}}$ consists of models satisfying:*

- (A0) *Initial hidden states are position-independent token embeddings: $h_i^{(0)} = E(x_i)$, with no absolute positional embeddings, index features, counters, or other access to i. The only positional dependence in the model is the distance-bias term in* (A4)*.*
- (A1) *The model is a depth-D causal transformer-style model with arbitrary width and skip connections as in* (1)*.*
- (A2) *Execution is deterministic p-bit execution as in Appendix D, including the left-to-right softmax accumulation described above.*
- (A3) *At layer ℓ and head h, queries, keys, and values are functions only of layer-$(\ell-1)$ hidden states.*
- (A4) *Positional information enters only as a distance-dependent additive logit bias:*
$$e_{i,j}^{(\ell,h)} = \mathsf{fl}\big((q_i^{(\ell,h)})^\top k_j^{(\ell,h)} - b_h^{(\ell)}(i-j)\big).$$
- (A5) *Each $b_h^{(\ell)}$ is eventually constant: there is $n_h^{(\ell)}$ such that $b_h^{(\ell)}(d+1) = b_h^{(\ell)}(d)$ for all $d \ge n_h^{(\ell)}$. We write $\bar{n}_\ell := \max_h n_h^{(\ell)}$.*
- (A6) *The readout is deterministic and position-free from the final hidden state.*

D.2.1 TERMINAL-RUN SATURATION AND THE COPYING LOWER BOUND

Throughout the lower-bound proof, fix a prefix x and a token σ. We compare the terminal-run contexts $R_{x,k}(\sigma)$ and $R_{x,k+1}(\sigma)$ for the same prefix. The prefix may still affect the logits; the claim is only that, after fixing x, a sufficiently long terminal σ-run has length-invisible next-token logits. The following layerwise statement proves the terminal-run saturation lemma in the main text (Lemma 3.3).

Lemma D.6 (Layerwise terminal-run saturation). *Fix $x \in \Gamma^*$ and $\sigma \in \mathcal{V}$. Define*

$$\kappa_0 := 0, \qquad \kappa_\ell := \kappa_{\ell-1} + \bar{n}_\ell + 2^p.$$

For every layer ℓ, there is a state $u_{x,\sigma}^{(\ell)}$ such that

$$H_{x,\sigma}^{(\ell)}(j) = u_{x,\sigma}^{(\ell)}$$

for all $j \geq \kappa_\ell$ with $j \geq 1$. Consequently, for $k, k' \geq \kappa_D$,

$$h_{\text{last}}^{(D)}(R_{x,k}(\sigma)) = h_{\text{last}}^{(D)}(R_{x,k'}(\sigma))$$

bit-for-bit. Hence

$$\text{Logits}_N(R_{x,k}(\sigma)) = \text{Logits}_N(R_{x,k'}(\sigma)).$$

Proof. We induct on ℓ. The case $\ell = 0$ follows from position-independent token embeddings, since every terminal-run token is σ. Assume the claim at layer $\ell - 1$, and consider a terminal-run query $\widehat{s}_j$ with

$$j \geq \kappa_\ell = \kappa_{\ell-1} + \bar{n}_\ell + 2^p.$$

By the induction hypothesis and causality, all terminal-run tokens with index at least $\max\{1, \kappa_{\ell-1}\}$ have the same previous-layer state, hence the same query, key, and value. For any head h, split the causal scan of $\widehat{s}_j$ into

$$\text{fixed prefix and pre-plateau run} \quad | \quad \widehat{s}_{\max\{1,\kappa_{\ell-1}\}}, \ldots, \widehat{s}_{j-n_h^{(\ell)}} \quad | \quad \widehat{s}_{j-n_h^{(\ell)}+1}, \ldots, \widehat{s}_j.$$

The last part is empty if $n_h^{(\ell)} = 0$. The first part is independent of j: its hidden states are fixed by causality, and its distances from $\widehat{s}_j$ are already beyond the eventual-constancy threshold. The middle part contains at least 2^p identical key/value/logit contributions. The last part has bounded length and is a fixed relative-distance list. The max-shift is therefore independent of j. By constant-block invariance, changing the length of the middle block does not change the accumulator state after leaving it. Thus every head output, and hence the layer-ℓ hidden state after the deterministic layer update, is independent of j. This proves the induction. The final hidden-state equality follows by taking $\ell = D$, and the logit equality follows from the position-free deterministic readout (A6). □

Proof of Theorem 3.4. We show that a correct all-identical copy cannot be followed by correct termination once $\kappa_D = o(N)$. Since $\kappa_D = o(N)$, for all sufficiently large N and every $L \in [N/3, N/2]$ we have $L - 1 \geq \kappa_D$. Let

$$x_L := \langle \text{BOS} \rangle, \sigma^L, \langle \text{EOI} \rangle, \langle \text{COPY} \rangle.$$

For any model claimed to solve $\text{Copy}_{\leq N}$, the contexts up to the required termination query $R_{x_L,L}(\sigma)$ are necessarily in its domain. If the first L generated tokens are not all σ, the model has already failed. Otherwise, the L-th token σ was generated from $R_{x_L,L-1}(\sigma)$. By Lemma D.6, all contexts $R_{x_L,k}(\sigma)$ with $k \geq L - 1$ have identical logits. Deterministic greedy decoding therefore outputs σ at $R_{x_L,L}(\sigma)$, where the correct token is $\langle \text{EOS} \rangle$, and then continues outputting σ for as long as the context remains in the model domain. □

Definition D.7 (ALiBi and Hard-ALiBi distance biases). *For a query position i and key position $j \leq i$, write $d = i - j$. In our sign convention, the attention logit is $q_i^\top k_j - b_h(d)$.*

Following Press et al. (2022), an ALiBi *head with slope $m_h \geq 0$ uses the distance penalty*

$$b_h^{\text{ALiBi}}(d) = m_h d.$$

This exact-real bias is not eventually constant when $m_h > 0$; the same is true if one computes $m_h d$ directly and rounds each value independently. The recursive implementation of ALiBi mentioned in Corollary 3.5 is computed by the rounded recurrence

$$b_h(0) = 0, \qquad b_h(d+1) = \mathsf{fl}(b_h(d) + m_h).$$

Following Jelassi et al. (2024), an ideal Hard-ALiBi *head with window length $w_h \in \mathbb{N}$ masks all positions outside the window:*

$$b_h^{\text{H-ALiBi}}(d) = \begin{cases} 0, & 0 \le d < w_h, \\ +\infty, & d \ge w_h. \end{cases}$$

Thus the head has an unpenalized causal window of length w_h. An implementation that fits our finite-precision model can either remove the masked positions from the softmax, equivalently making their shifted weights exactly zero, or replace $+\infty$ by a representable penalty B_h large enough that every window-external shifted logit is at most $-\mathcal{M}_p$. The latter requirement depends on a bound for the content-logit gap; without such a bound, a finite B_h is only a large finite penalty, not an exact hard mask. Unmasked Hard-ALiBi heads are NoPE heads, $b_h \equiv 0$.

Bias thresholds for the corollary. To apply the lower bound to the three cases in Corollary 3.5, it remains to bound their eventual-constancy indices.

Lemma D.8 (Recursive implementation of ALiBi is eventually constant)**.** *Let $m_h \ge 0$ be a p-bit float and define $b_h(0) = 0$, $b_h(d+1) = \mathsf{fl}(b_h(d) + m_h)$. Assuming finite intermediates do not overflow, there exists $n_h \le 2^p$ such that $b_h(d+1) = b_h(d)$ for all $d \ge n_h$.*

Proof. If $m_h \ge 0$ and x is finite and representable, then $x + m_h \ge x$ and round-to-nearest-even cannot round the result to a representable value below x. Hence $b_h(d+1) = \mathsf{fl}(b_h(d) + m_h) \ge b_h(d)$. $(b_h(d))_{d \ge 0}$ can make at most 2^p strict increases among p-bit states, so it becomes constant after at most 2^p distinct representable values. Thus there exists $n_h \le 2^p$ such that $b_h(d+1) = b_h(d)$ for all $d \ge n_h$. □

Lemma D.9 (Hard-ALiBi is eventually constant)**.** *A finite-window Hard-ALiBi head with window length w_h as in Definition D.7 is constant outside the window: either the masked contribution is zero in the ideal hard-mask implementation, or $b_h(d+1) = b_h(d)$ for all $d \ge w_h$ in the finite-penalty implementation. Thus its eventual-constancy index is at most w_h.*

Proof. For every $d \ge w_h$, both d and $d+1$ fall outside the unpenalized window. In the ideal hard-mask version, both positions are removed from the softmax, equivalently both shifted weights are fixed at 0. In a finite-penalty version, both distances receive the same outside-window penalty B_h, so $b_h(d) = b_h(d+1) = B_h$. □

General Hard-ALiBi window condition. For a Hard-ALiBi family, let $\bar{w}_\ell := \max_h w_h^{(\ell)}$ over the hard-masked heads in layer ℓ, treating unmasked NoPE heads as having index 0. Theorem 3.4 applies whenever

$$\sum_{\ell=1}^{D} (\bar{w}_\ell + 2^p) = o(N).$$

The main-text corollary uses the simpler sufficient condition $\bar{w}_\ell \le 2^p$ for every layer.

Proof of Corollary 3.5. We bound κ_D by $2D\,2^p$ in each of the three cases, under the stated no-overflow assumption for recursive ALiBi. For NoPE, $b_h^{(\ell)} \equiv 0$ for every layer and head, so $\bar{n}_\ell = 0$ and $\kappa_D = D\,2^p$. For the recursive implementation of ALiBi with finite intermediates, Lemma D.8 gives $n_h^{(\ell)} \le 2^p$ for every head, so $\bar{n}_\ell \le 2^p$ and $\kappa_D \le 2D\,2^p$. For bounded-window Hard-ALiBi with all windows at most 2^p, Lemma D.9 gives $n_h^{(\ell)} \le 2^p$ for every hard-masked head; unmasked heads have $n_h^{(\ell)} = 0$. Hence again $\bar{n}_\ell \le 2^p$ and $\kappa_D \le 2D\,2^p$. Exact-real ALiBi $b(d) = md$ is outside this corollary unless it is replaced by the recursive implementation of ALiBi and becomes eventually constant. Since $D(N)2^{p(N)} = o(N)$,

$$\kappa_D \le 2D(N)\,2^{p(N)} = o(N).$$

Theorem 3.4 applies in all cases. In particular, $D = N^{0.01}$ and $p \le 0.9 \log_2 N$ give $D\,2^p \le N^{0.91} = o(N)$. If $p = \log_2 N - C \log\log N$, then the same sufficient condition requires $D = o((\log N)^C)$. □

D.3 Formal VO Construction with CRT Periods

This subsection gives the fully formal instantiation of the main-body VO-rotated head in Definition 3.6 and the class $\mathcal{T}_{\mathrm{VO}}$ in Definition 3.7. Throughout this subsection, R_q denotes the fixed two-dimensional period-q rotation

$$R_q := \begin{pmatrix} \cos(2\pi/q) & -\sin(2\pi/q) \\ \sin(2\pi/q) & \cos(2\pi/q) \end{pmatrix},$$

and $\mathsf{R}_{q,m} := \mathrm{diag}(R_q, \dots, R_q)$ denotes the even-dimensional rotary action. Powers of both R_q and $\mathsf{R}_{q,m}$ are interpreted modulo q. Write $\rho_q(r) := R_q^r \mathbf{e}_1$ for the ideal rotary vector and $\widehat{R}_q(r) := \mathsf{fl}(R_q^r)$ for the rotation table, with exact index $r \bmod q$ and entrywise rounding. Let $\mathsf{mv}_p(B, \mathbf{v})$ denote a matrix-vector product with each multiplication and addition rounded to p bits. For the prime periods used below, let $\Phi_h(t)$ be the computed Layer-1 rotary output and $\Phi(t) := (\Phi_h(t))_{h=1}^{H}$. Its ideal counterpart is $\Phi_h^*(t) := \rho_{q_h}(\delta(t))$; Lemma D.14 bounds the error between them.

The construction below specifies the positional encoding and both layers before verifying their routing and precision guarantees.

Definition D.10 (VO-rotated attention head)**.** *We use the VO head of Definition 3.6, with integer period $q_h \ge 2$ and even rotary value dimension m_h. Its attention weights use ordinary unrotated query/key logits; only the rotary value subspace undergoes the position-indexed pre- and post-rotations. In p-bit execution, these rotations use exact table indices and rounded arithmetic as specified below.*

Definition D.11 (Prime-q_h VO-anchor head)**.** *A* prime-q_h VO-anchor head *is a NoPE causal attention head with one-dimensional query/key coordinates, ordinary logits $e_{i,j} = \mathsf{fl}(q_i k_j)$, a two-dimensional rotary value subspace, and prime period q_h. Before rounding, its rotary head-local pre-output is*

$$\widetilde{\mathbf{o}}_t^{\mathrm{rot}} = R_{q_h}^t \sum_{j \le t} \alpha_{t,j} R_{q_h}^{-j} \mathbf{v}_j^{\mathrm{rot}},$$

where powers of the fixed two-dimensional rotation R_{q_h} are interpreted modulo q_h. If this head attends one-hot to position j and $\mathbf{v}_j^{\mathrm{rot}} = \mathbf{e}_1$, its ideal rotary output is $\rho_{q_h}(t-j)$; the computed output uses the rounded rotation tables.

Definition D.12 (Formal $\mathcal{T}_{\mathrm{VO}}$ instantiation with CRT periods)**.** *For the construction theorem, $\mathcal{T}_{\mathrm{VO}}$ is instantiated with prime periods as follows. Clause* (B5) *specifies the positional primitive; the other clauses choose the two layers, resource parameters, and token readout for this construction.*

(B1) *Layer 1 has H prime-q_h VO-anchor heads for the first H primes, each with $m_h = 2$. The output projection writes their $2H$ head-local pre-output coordinates into a fingerprint subspace and leaves the token-embedding subspace untouched;* $\mathrm{MLP}^{(1)} = 0$.

(B2) *Layer 2 is a single readout attention head whose query and key vectors are partitioned into H prime blocks. The index h in $\mathbf{q}_t^{(2,h)}$ and $\mathbf{k}_j^{(2,h)}$ denotes a block within this head. Its logit is the sequential p-bit sum*

$$E_{t,j}^{(2)} = \sum_{h=1}^{H}{}^{\mathsf{fl}} \langle \mathbf{q}_t^{(2,h)}, \mathbf{k}_j^{(2,h)} \rangle_p. \tag{2}$$

Here $\sum^{\mathsf{fl}}$ denotes left-to-right rounded summation, and $\langle \cdot, \cdot \rangle_p$ denotes an inner product evaluated in p-bit arithmetic. Its value vector is the preserved token embedding $\mathbf{v}_j^{(2)} = \mathbf{x}_j^{(0)}$, and its output projection writes the returned value into the token-readout block used by the final decoder.

(B3) $H = \Theta(\log N / \log\log N)$, *so* $q_{\max} = \Theta(\log N)$ *and* $\Pi := \prod_h q_h > N + 1$.

(B4) $p_m = c_m \lceil \log_2 \log_2 \Pi \rceil$ *with* $c_m \ge 4$ *and* $p_e = c_e \lceil \log_2 \log_2 \Pi \rceil$ *with c_e a sufficiently large absolute constant.*

(B5) *Each position has exact access to its index modulo q_h, which selects the corresponding rounded rotation-table entry. The construction uses $m_h = 2$, so its value rotations apply $\widehat{R}_{q_h}(r)$ using mv_p. Implementation and storage costs are given in Section D.4.4.*

(B6) *For any fixed finite vocabulary $\mathcal{V}$, choose $d_{\mathrm{tok}} = d_{\mathrm{tok}}(|\mathcal{V}|)$, token embeddings for $\mathcal{V} \cup \{\langle \textsc{BOS}\rangle, \langle \textsc{EOI}\rangle, \langle \textsc{COPY}\rangle, \langle \textsc{EOS}\rangle\}$, a token-readout block, and a fixed linear readout. The greedy argmax recovers vocabulary tokens from the returned token vector and maps the returned $\langle \textsc{EOI}\rangle$ embedding to $\langle \textsc{EOS}\rangle$. One may take one-hot token embeddings with identity rows for $\mathcal{V}$, an $\langle \textsc{EOS}\rangle$ row that scores the $\langle \textsc{EOI}\rangle$ coordinate, and zero readout weights on the other coordinates.*

Layer-1 instantiation. For each prime head h,

$$(q_t^{(1)}, k_t^{(1)}) = \begin{cases} (0, -\mathcal{M}_p), & t = \langle \textsc{BOS}\rangle, \\ (1, 0), & t = \langle \textsc{COPY}\rangle, \\ (1, -2\mathcal{M}_p), & \text{otherwise}, \end{cases} \tag{3}$$

with value vector $\mathbf{v}_t^{(1)} = \mathbf{e}_1$ in the two-dimensional rotary subspace. The resulting computed component $\Phi_h(t)$ approximates $\Phi_h^*(t)$ through two rounded rotations, as shown in Lemma D.14.

Layer-2 instantiation. Let C_0 be a universal constant large enough to cover the Layer-1 rotation error, the rounded one-step query rotation, and the finite-precision multiplications and additions inside each two-dimensional inner product. The constant C_0 is independent of N, H, q_h, and the input length. It may depend only on fixed local arithmetic-format constants and constant-dimensional block operations. Let C_Σ be a universal constant large enough to absorb both the left-to-right summation error from Lemma D.17 and the aggregate per-block rounding errors. Using the fingerprint coordinates written by Layer 1, for each prime block h:

$$\mathbf{q}_t^{(2,h)} = \big[A\,\mathsf{mv}_p(\widehat{R}_{q_h}(1), \Phi_h(t)),\ -A^2\big], \qquad \mathbf{k}_j^{(2,h)} = \big[A\,\Phi_h(j),\ 1\big], \tag{4}$$

where $\widehat{R}_{q_h}(1)$ is the rounded one-step rotation for period q_h. With ideal fingerprints and exact arithmetic, the centered dot product is

$$A^2\left(\langle R_{q_h}\Phi_h^*(t), \Phi_h^*(j)\rangle - 1\right).$$

A perfect match for offset $k+1$ gives ideal centered inner product 0. Any wrong position mismatches in at least one prime coordinate, producing a negative cosine gap. Scaling by A^2, with rounding error controlled by the choice of precision, pushes all wrong logits below the softmax underflow threshold. Here η budgets the rounding error before scaling, γ is the remaining cosine gap, and A^2 amplifies this gap to the softmax underflow scale:

$$\eta := C_\Sigma H^2 \varepsilon_{\mathrm{mach}}, \qquad \gamma := 8/q_{\max}^2 - \eta, \qquad A := 2^{\lceil \frac{1}{2}\log_2(4\mathcal{M}_p/\gamma)\rceil}. \tag{5}$$

With $p_m = c_m\lceil \log_2 \log_2 \Pi\rceil$ and $c_m \geq 4$, we have $\eta = o(1/q_{\max}^2)$, so $\gamma > 0$ for all sufficiently large N. Since A is a power of two, both A and A^2 are exactly representable binary floating-point numbers whenever the exponent range contains their exponents. Throughout the Layer-2 proof, take N large enough that $\eta \leq 2/q_{\max}^2$. Then $\gamma \geq 6/q_{\max}^2 > 0$ and $\eta \leq \gamma/3$.

D.4 Proofs for the VO Construction with CRT Periods

We first verify anchor selection, then bound the error in the stored fingerprints and the Layer-2 logits.

D.4.1 Layer-1 anchor selection and fingerprints

Lemma D.13 (Layer-1 exact one-hot)**.** *Under (3), Layer 1 satisfies $\alpha_{t,a(t)}^{(1)} = 1$ bit-for-bit for every non-$\langle \textsc{BOS}\rangle$ position.*

Proof. For every non-$\langle \textsc{BOS}\rangle$ query position, $q_t^{(1)} = 1$, so the ordinary attention logit to candidate j is simply $k_j^{(1)}$.

If $\langle$COPY$\rangle$ has not appeared in the causal prefix up to t, the visible $\langle$BOS$\rangle$ key is $-\mathcal{M}_p$, while every other visible key is $-2\mathcal{M}_p$. Hence the $\langle$BOS$\rangle$ position $a(t)$ is the unique maximizer. After max-shift, the $\langle$BOS$\rangle$ weight is $\mathrm{fl}(\exp(0)) = 1$, and every other shifted logit is at most $-\mathcal{M}_p$, so its exponential rounds to 0 by Definition D.1.

If $\langle$COPY$\rangle$ has appeared, the unique $\langle$COPY$\rangle$ position has key 0, $\langle$BOS$\rangle$ has key $-\mathcal{M}_p$, and all other visible positions have key $-2\mathcal{M}_p$. Thus the $\langle$COPY$\rangle$ position $a(t)$ is the unique maximizer. Again the winner contributes weight 1 and every non-winner is at least $\mathcal{M}_p$ below the maximum after max-shift, so all non-winner weights round to 0. In both cases the denominator is exactly 1 and the attention is bit-for-bit one-hot on $a(t)$. □

With this one-hot attention, the denominator is 1 and all non-anchor numerator contributions are zero. Pre-rotation of $\mathbf{e}_1$ selects the first column of the rounded table exactly, and accumulation preserves that column. The computed rotary output is therefore

$$\Phi_h(t) = \mathsf{mv}_p\Big(\widehat{R}_{q_h}(t), \widehat{R}_{q_h}(-a(t))\mathbf{e}_1\Big). \tag{6}$$

The same expression holds at $\langle$BOS$\rangle$, whose causal attention contains only itself.

Lemma D.14 (Fingerprint). *For all sufficiently large N, the computed Layer-1 outputs satisfy:*

(i) $\Phi_h(t)$ *depends only on the pair* $(t \bmod q_h, a(t) \bmod q_h)$.

(ii) $\|\Phi_h(t) - \rho_{q_h}(\delta(t))\|_\infty \le C_{\mathrm{rot}}\varepsilon_{\mathrm{mach}}$, *where* C_{rot} *is universal; one may take* $C_{\mathrm{rot}} = 8$.

(iii) $\Phi(t)$ *uniquely identifies* $\delta(t)$ *for* $\delta(t) \le N + 1 < \Pi$.

Proof. Part (i) follows from (6) and exact table indexing. For part (ii), fix a head, abbreviate $q = q_h$ and $a = a(t)$, and put $v = R_q^{-a}\mathbf{e}_1$ and $\widehat{v} = \widehat{R}_q(-a)\mathbf{e}_1$. Every rounded table entry has magnitude at most 1 and absolute error at most $\varepsilon_{\mathrm{mach}}$. Thus $\|\widehat{v} - v\|_\infty \le \varepsilon_{\mathrm{mach}}$ and $\|\widehat{v}\|_\infty \le 1$. Using the matrix row-sum norm and $\|R_q^t\|_\infty \le \sqrt{2}$, we obtain

$$\|\widehat{R}_q(t)\widehat{v} - R_q^t v\|_\infty \le (2 + \sqrt{2})\varepsilon_{\mathrm{mach}}.$$

Each coordinate of $\mathsf{mv}_p(\widehat{R}_q(t), \widehat{v})$ requires two products of magnitude at most 1 and one sum of magnitude at most 2. For sufficiently large N, rounding these operations introduces at most $\varepsilon_{\mathrm{mach}}$ per product and $2\varepsilon_{\mathrm{mach}}$ for the sum, including underflow; the exponent range in (B4) contains these bounded intermediates without overflow. The arithmetic error is therefore at most $4\varepsilon_{\mathrm{mach}}$ per coordinate. Since $R_q^t v = \rho_q(t - a)$ in exact arithmetic, the total error is at most $(6 + \sqrt{2})\varepsilon_{\mathrm{mach}} < 8\varepsilon_{\mathrm{mach}}$, independently of t, a, and q.

For part (iii), two distinct offsets $\delta, \delta' \in \{0, \dots, N + 1\}$ differ modulo some prime q_h by CRT, since $|\delta - \delta'| < \Pi$. For their distinct residues r, r' in this head,

$$\|\rho_{q_h}(r) - \rho_{q_h}(r')\|_2^2 = 2\big(1 - \cos(2\pi(r - r')/q_h)\big) \ge 16/q_h^2$$

by Fact D.15. Hence the ℓ_∞ separation is at least $4/(q_h\sqrt{2}) > 2/q_h$. The two computation errors subtract at most $2C_{\mathrm{rot}}\varepsilon_{\mathrm{mach}}$, which is smaller for sufficiently large N, since $c_m \ge 4$ and $q_h \le q_{\max} = \Theta(\log N)$. Thus computed fingerprints belonging to distinct offsets remain distinct, even when their absolute positions and anchors differ. □

The Layer-2 margin argument needs only that different residues are separated by a cosine gap. The closest two different residues are one step apart on a period-n circle.

Fact D.15 (Cosine gap). *For integer $n \ge 2$, $1 - \cos(2\pi/n) \ge 8/n^2$.*

Proof. For $\theta \in [0, \pi]$, $1 - \cos\theta = 2\sin^2(\theta/2)$, and $\sin x \ge 2x/\pi$ on $[0, \pi/2]$. Thus $1 - \cos\theta \ge 2\theta^2/\pi^2$. Substituting $\theta = 2\pi/n$ gives $1 - \cos(2\pi/n) \ge 8/n^2$. □

D.4.2 LAYER-2 MATCHING AND FINITE PRECISION

For one prime block, the centered inner product is near zero when the shifted query residue matches the key residue. A mismatch costs at least the cosine gap, up to rounding error.

Lemma D.16 (Per-prime-block logit bounds)**.** *Let $r_q = (\delta(t)+1) \bmod q_h$ and $r_k = \delta(j) \bmod q_h$. For all sufficiently large N, we have*

$$\langle \mathbf{q}_t^{(2,h)}, \mathbf{k}_j^{(2,h)} \rangle_p = A^2\big(\cos(2\pi(r_q - r_k)/q_h) - 1\big) + \xi_h, \qquad |\xi_h| \le C_0 A^2 \varepsilon_{\mathrm{mach}}.$$

In particular, matching residues give an absolute value at most $C_0 A^2 \varepsilon_{\mathrm{mach}}$, while mismatched residues give

$$\langle \mathbf{q}_t^{(2,h)}, \mathbf{k}_j^{(2,h)} \rangle_p \le -A^2\big(8/q_h^2 - C_0\varepsilon_{\mathrm{mach}}\big).$$

Proof. Write $\tilde{u} := \mathsf{mv}_p(\widehat{R}_{q_h}(1), \Phi_h(t))$ and $\tilde{v} := \Phi_h(j)$. Lemma D.14 gives a uniform $C_{\mathrm{rot}}\varepsilon_{\mathrm{mach}}$ error for each stored component. The additional one-step query rotation has bounded entries and a fixed number of rounded operations, so it adds only $O(\varepsilon_{\mathrm{mach}})$ error. Consequently,

$$\tilde{u} = \rho_{q_h}(r_q) + O(\varepsilon_{\mathrm{mach}}), \qquad \tilde{v} = \rho_{q_h}(r_k) + O(\varepsilon_{\mathrm{mach}}).$$

The exact inner product of the unrounded unit vectors is

$$\langle \rho_{q_h}(r_q), \rho_{q_h}(r_k) \rangle = \cos(2\pi(r_q - r_k)/q_h).$$

Scaling by A, multiplying coordinates, summing the two coordinates, and adding the centering term $-A^2$ introduce only $O(A^2\varepsilon_{\mathrm{mach}})$ error. The universal constant C_0 absorbs the Layer-1 errors from Lemma D.14, the rounded one-step query rotation, scaling by the representable power of two A, the two rounded products, and the rounded addition and centering by $-A^2$. If $r_q = r_k$, the exact centered value is 0. If $r_q \neq r_k$, Fact D.15 gives $\cos(2\pi(r_q - r_k)/q_h) - 1 \le -8/q_h^2$, yielding the stated upper bound after rounding. □

To combine the per-block bounds, we must also control rounding in the sum over prime blocks.

Lemma D.17 (Left-to-right summation error)**.** *Let $z_1, \ldots, z_H$ be finite p-bit numbers with $|z_h| \le C_A A^2$. Assume no overflow occurs during the deterministic left-to-right p-bit summation and $H\varepsilon_{\mathrm{mach}}$ is sufficiently small. Then, for a constant C_Σ depending only on C_A and fixed arithmetic-format constants,*

$$\left| \sum_{h=1}^{H}{}^{\mathsf{fl}} z_h - \sum_{h=1}^{H} z_h \right| \le C_\Sigma H^2 A^2 \varepsilon_{\mathrm{mach}}.$$

Proof. Let $\widehat{S}_m$ be the rounded left-to-right partial sum and $S_m = \sum_{h=1}^{m} z_h$ the exact real partial sum. Inductively, as long as $H\varepsilon_{\mathrm{mach}}$ is small, $|\widehat{S}_m| \le C'_A H A^2$ for a constant C'_A depending only on C_A. Thus the m-th rounded addition has absolute rounding error at most $C''_A H A^2 \varepsilon_{\mathrm{mach}}$. Summing this bound over the H additions gives

$$|\widehat{S}_H - S_H| \le C_\Sigma H^2 A^2 \varepsilon_{\mathrm{mach}},$$

after increasing C_Σ by a fixed constant factor. □

At a copy-region query of offset k, all visible copy-region offsets are at most k. The source side contains exactly one position of offset $k+1$ for $0 \le k \le L$, so the target is unique among visible positions. The next lemma compares its logit with those of all other visible positions.

Lemma D.18 (Layer-2 logit bounds)**.** *Let copy position t have offset k, and let $j^\star$ be the unique position with $\delta(j^\star) = k+1$. For all sufficiently large N, provided $k+1 \le N+1 < \Pi$:*

(i) $|E^{(2)}_{t,j^\star}| \le A^2\eta$.

(ii) $E^{(2)}_{t,j} \le -A^2\gamma$ *for every* $j \neq j^\star$ *with* $j \le t$.

Proof. For the correct match $j^\star$, every prime block has $\delta(j^\star) \equiv k+1 \pmod{q_h}$, so the residues match at every prime. Lemma D.16 bounds the exact sum of the H rounded block contributions by $O(HA^2\varepsilon_{\mathrm{mach}})$ in absolute value. Lemma D.17, with C_Σ chosen large enough to absorb both errors, then gives part (i).

For a wrong position $j \neq j^\star$, CRT uniqueness implies that at least one prime $h^\star$ has

$$\delta(j) \not\equiv k+1 \pmod{q_{h^\star}}.$$

For this mismatched block, Lemma D.16 gives

$$\langle \mathbf{q}_t^{(2,h^\star)}, \mathbf{k}_j^{(2,h^\star)} \rangle_p \leq -A^2(8/q_{\max}^2 - C_0\varepsilon_{\mathrm{mach}}).$$

Every matched block can contribute at most $+C_0A^2\varepsilon_{\mathrm{mach}}$, and every other mismatched block can only decrease the upper bound for all sufficiently large N. By the standing Layer-2 condition after (5), $\gamma \geq 6/q_{\max}^2 > 0$, so the wrong position has a negative gap of order $A^2/q_{\max}^2$. Summing over all prime blocks and applying Lemma D.17, with C_Σ absorbing the aggregate per-block errors,

$$E_{t,j}^{(2)} \leq -A^2\big(8/q_{\max}^2 - \eta\big) = -A^2\gamma.$$

□

Once the wrong logits are at least $\mathcal{M}_p$ below the maximum, their exponentials underflow to zero. The softmax is therefore exactly one-hot on the matching offset.

Lemma D.19 (Layer-2 exact one-hot)**.** *Let copy position t have offset k, and let $j^\star$ be the unique visible position with $\delta(j^\star) = k+1$. Provided $k+1 \leq N+1 < \Pi$, Layer 2 satisfies $\alpha_{t,j^\star}^{(2)} = 1$ bit-for-bit.*

Proof. Lemma D.18 shows that $j^\star$ achieves the maximum logit. After max-shift, its shifted logit is 0 and its weight is $\mathrm{fl}(\exp(0)) = 1$. For every wrong position,

$$E_{t,j}^{(2)} - E_{t,j^\star}^{(2)} \leq -A^2\gamma + A^2\eta \leq -(2/3)A^2\gamma \leq -\mathcal{M}_p,$$

using the standing Layer-2 condition $\eta \leq \gamma/3$ and the power-of-two choice $A^2 \geq 4\mathcal{M}_p/\gamma$. Thus every wrong exponential underflows to 0, the softmax denominator is 1, and $\alpha_{t,j^\star}^{(2)} = 1$ bit-for-bit. □

D.4.3 Copying and resource bounds

Lemma D.20 (VO copying lemma with CRT periods)**.** *For the formal instantiation in Definition D.12:*

(i) *Layer 1 attends bit-for-bit one-hot to the active anchor $a(t)$.*

(ii) *The H VO outputs form a CRT fingerprint that uniquely determines $\delta(t)$ whenever $\delta(t) \leq N+1 < \Pi$.*

(iii) *At any copy-region query of offset k, Layer 2 attends bit-for-bit one-hot to the unique visible position with offset $k+1$.*

Proof of Lemma D.20. Part (i) is Lemma D.13. Part (ii) is Lemma D.14. For part (iii), at a copy-region query of offset k, the visible position with offset $k+1$ is unique whenever $k+1 \leq N+1 < \Pi$; Lemma D.19 then gives exact one-hot attention to that position. □

Proof of Theorem 3.8. We use the copying lemma to obtain the required outputs, then account for precision and dimension. Fix a sufficiently large design horizon N, a finite vocabulary $\mathcal{V}$, arbitrary $L \leq N$, and arbitrary source tokens $s_1, \ldots, s_L \in \mathcal{V}$. At a copy-region query t of offset $0 \leq k \leq L$, Lemma D.20(iii) gives bit-for-bit one-hot attention to the unique visible position $j^\star$ of offset $k+1$. This position holds s_{k+1} if $k < L$ and $\langle \mathrm{EOI} \rangle$ if $k = L$. Since $W_O^{(1)}$ writes only to the fingerprint subspace and $\mathrm{MLP}^{(1)} = 0$, the Layer-1 skip connection preserves the token embedding $\mathbf{x}_{j^\star}^{(0)}$. Layer 2 writes its returned value into the token-readout block, so the fixed linear readout from (B6) and greedy argmax output the correct vocabulary token, or output $\langle \mathrm{EOS} \rangle$ when the returned token is $\langle \mathrm{EOI} \rangle$.

It remains to account for resources. The first H primes satisfy $q_{\max} = \Theta(H \log H)$ and $\prod_{h \le H} q_h > N + 1$ for $H = \Theta(\log N / \log\log N)$, by the prime number theorem. Hence $q_{\max} = \Theta(\log N)$. With $p_m = c_m \lceil \log_2 \log_2 \Pi \rceil$ and $c_m \ge 4$,

$$H^2 \varepsilon_{\mathrm{mach}} = O\left(\frac{(\log N)^2}{(\log\log N)^2}\right)(\log N)^{-c_m} = o(1/\log^2 N),$$

so $\eta = o(1/q_{\max}^2)$, $\gamma > 0$, and the cosine gap survives rounding. The exponent in $A = 2^{\lceil \frac{1}{2} \log_2(4\mathcal{M}_p/\gamma) \rceil}$ is $O(\log\log N)$ because $\mathcal{M}_p = \Theta(2^{p_e})$ and $\gamma^{-1} = (\log N)^{O(1)}$. Hence a sufficiently large constant c_e in $p_e = c_e \lceil \log_2 \log_2 \Pi \rceil$ gives enough exponent range to represent A and A^2 exactly. Since $A^2 \ge 4\mathcal{M}_p/\gamma$ by construction, the underflow inequalities above remain valid. Thus all calculations above use $p = O(\log\log N)$ bits. The added fingerprint subspace has dimension $2H = \Theta(\log N / \log\log N)$. The total residual dimension can be taken as $d = d_{\mathrm{tok}}(|\mathcal{V}|) + 2H + 1$. As in (B5), exact residues select the rounded rotation tables; Lemma D.14 and the Layer-2 bounds account for all subsequent rotation errors. □

D.4.4 Implementation and resource conventions

Implementing (B5). Exact modular reduction modulo each chosen period selects the rotation-table indices, as required by Definition 3.7.

What is assumed exact is the *selection* of the phase, that is, the residue $t \bmod q_h$. The phase values and rotation arithmetic are rounded to p bits: Lemma D.14 bounds the total Layer-1 error by $C_{\mathrm{rot}} \varepsilon_{\mathrm{mach}}$ for a universal constant C_{rot}, and the Layer-2 gap analysis in (5) carries it through to the logits. The primitive can be supplied by phase tables indexed by exact residues: head h needs q_h sine/cosine pairs, giving $O(\log^2 N / \log\log N)$ stored scalars across heads, each rounded to p bits. A sequential position generator supplies these indices with a cyclic counter $c_{h,0} = 0$, $c_{h,t+1} = (c_{h,t} + 1) \bmod q_h$ for each head. Since $q_h = O(\log N)$, each counter uses $O(\log\log N)$ bits, within the construction's precision scale; total counter state is $O(\log N)$ bits across heads. The counters index the rounded phase tables and depend only on position, while the attention computation selects the active anchor and forms its relative offset. Table storage and counter state are accounted for separately from the precision of each stored coordinate. This separation of phase generation from activation arithmetic is natural for rotary PEs: for example, the Hugging Face Llama implementation computes RoPE phases and trigonometric values in float32 before casting them to the activation dtype.[3]

In exact arithmetic, both RoPE and VO use the composition identity $R^t R^{-j} = R^{t-j}$, but in different computations: RoPE uses rotations in query/key inner products, whereas this construction returns offset representations for the next layer to match. The VO guarantee requires distinct residues to remain distinguishable after rounding. Small periods keep the cosine gap in Fact D.15 at least $8/q^2$, which the Layer-2 precision analysis controls.

Remark on head-wise periods. Definition D.11 assigns one period per head, but distributing the residues across heads is a presentational choice, not a resource requirement. Allowing different periods in different two-dimensional blocks of a single head implements the same computation: every prime head attends one-hot to the same active anchor, so a single head with the block-diagonal rotation $\mathrm{diag}(R_{q_1}, \ldots, R_{q_H})$ reproduces all H rotary outputs under the shared attention pattern, and Layer 2 already combines all prime blocks inside a single head via (2). The resources can therefore equivalently be counted as a total rotary dimension of $2H$ rather than a head count. One may also instantiate each prime head with any larger even m_h by placing $\mathbf{e}_1$ in the first block and leaving all other rotary coordinates unused; the resource accounting above uses $m_h = 2$.

[3] Transformers v4.44.2, `LlamaRotaryEmbedding.forward`.