QUANTITATIVE GENERALIZATION BOUNDS FOR MULTI-LAYER TRANSFORMERS: A FEW LONG STRINGS CAN BEAT MANY SHORT ONES

Anonymous authors
Paper under double-blind review

ABSTRACT

Transformer-based language models exhibit an astonishing capacity to generalize to longer and more difficult problem instances than those seen in training. However, theoretical understanding of this phenomenon still lags behind. Unlike most machine learning models, the generalisation abilities of transformers inherently require considering two axes. On one hand, one must examine in-distribution *statistical generalisation*, which concerns the ability of a model to broaden a function from training samples to novel samples of the same distribution. On the other hand, given the sequential nature of language data, one must also consider *length generalisation*, which reflects whether a model trained on inputs of a given length can extrapolate to longer ones. These two notions have mostly been studied separately, leaving open how they interact. In this work, we give a joint treatment of both axes for fixed-precision transformers. We provide bounds for generalisation guarantees in terms of the parameter sizes of the involved models, as well as the quantization space and the alphabet size that supply the sequences. Finally, we illustrate how the requirements interact between number and length of training samples. In particular, we find that increasing training length can be exponentially more efficient than increasing the sample size. Experiments support predictions of the theory.

1 INTRODUCTION

The success of large language models (LLMs) has prompted research into the learning and generalization properties of the underlying transformer architecture. Classical statistical learning theory provides a wealth of techniques that bound the generalization error as a function of the data size and the complexity of the hypothesis class. Recent work has brought these to bear on transformers (Covering numbers: Edelman et al. (2022); Trauger & Tewari (2024), Bayesian methods: Lintilhac & Shaikh (2026); Rizvi-Martel et al. (2026), VC Dimension: Yang et al. (2026c)). These results primarily address the *i.i.d. regime* – for transformers, this in particular implies that the test inputs are drawn from the same length distribution as the training inputs.

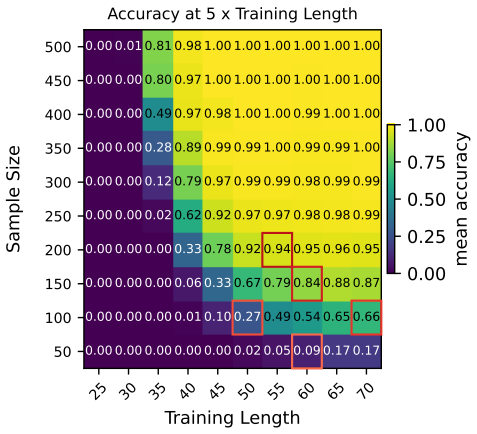


Figure 1: Our results predict that increasing the length of the training words can be much more token-efficient than increasing the number of samples: test accuracy for the Alphabet language A_{15} (5) on words five times longer than the training length. We highlight the best accuracy cells for the training token-budgets of $3000, 5000, 7000, 9000$, and 11000 tokens.

However, for transformer learning theory, the i.i.d. paradigm is arguably insufficient: it is highly desirable to understand when transformers can extrapolate to sequences *longer than seen during training*. This phenomenon is referred to as *length-generalization* (e.g. Bhattamishra et al., 2020; Anil et al., 2022; Kazemnejad et al., 2023), and is understood to be diagnostic of when a transformer actually learns an underlying algorithmic solution, as

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.

opposed to length-dependent heuristics (e.g. Zhou et al., 2024; Huang et al., 2026). It also indicates when LLMs can be expected to perform a task reliably across long contexts (Jobanputra et al., 2026).

Length generalization is governed by two quantities: the number of training samples, and their length. Sample count determines how reliably the training data pins down the target function, as in classical statistical learning theory; sample length determines how well the model extrapolates, since the longest input length at which the target function's behavior is observed dictates to which point a model will extrapolate beyond it. For a given target function and learning procedure, we thus need to consider not only its **sample complexity** (the required number of training samples), but also its **length complexity** (the required length of training samples).

Here, we take up the question of providing quantitative bounds on generalization *along both axes* for multi-layer transformers in the fixed-precision regime. We study the questions:

How do properties of a target function influence the length complexity and data complexity of the task?

When both sample size and length are bounded, how do they trade off? In particular, under a fixed token budget, how to best allocate sample size and length?

and address these by providing generalization bounds **depending on both sample size and input length**. The main idea is that training length determines which behaviors become observable; sample size determines how reliably those behaviors are learned. Below a task-dependent length threshold, some competing behaviors cannot be distinguished. Around the relevant transition, longer examples can make informative events much more common; increasing length can thus strongly reduce sample complexity in this regime. Beyond that transition, the benefit of additional length diminishes and additional independent examples matter more.

2 Background

2.1 Transformers

We study transformers in the **fixed-precision regime**, without positional encodings, as formalized by Li & Cotterell (2025; 2026) (see also Li et al. (2024); London & Kanade (2025)).

This class is expressive enough to accommodate key algorithmic primitives relevant to transformers, such as retrieving past symbols, checking global symbol alternation patterns, or the "induction head" construction. Fixed-precision transformers are less expressive than log- or unbounded-precision transformers; this turns out to be necessary for obtaining quantitative results, because length generalization bounds for the latter are known to be uncomputable in the worst case by results of Yang et al. (2026a). We discuss the detailed comparison across models of transformers in Appendix A.

Our model of transformers closely follows Li & Cotterell (2025): we consider transformers with softmax attention, fixed precision, no positional encodings and strict causal masking. However, we generalize over Li & Cotterell (2025) in viewing transformers not as binary classifiers, but as providing an output at every position. We provide details in Appendix B.2. All activations are computed in $\mathbb{F}$, a finite subset of $\mathbb{R}$ extended with $-\infty$ and ∞. The result of every operation is rounded to $\mathbb{F}$.

We fix and finite alphabet Σ. A *fixed-precision transformer* T, also just called *transformer* consists of token embeddings in dimension D, a sequence of L layers each consisting of H softmax attention heads with a ReLU feedforward network, and finally a linear output projection. Overall, it takes a word $w \in \Sigma^*$ input and returns vectors of dimension ν at every input position: $T(w) \in \mathbb{F}^{\nu \times |w|}$. We suppose that H always divides D, and that the dimension per head D_h is uniform among the heads. Assuming that $\Sigma, \mathbb{F}$ and the output dimension ν are fixed, we refer to these transformers as $\mathcal{FPT}(L, D, H)$. See Appendix B.2 for details.

2.2 Learning framework

We follow prior sample complexity bounds in studying an idealized learning model that abstracts away from the algorithmic process of learning. Capacity-based learning bounds (Edelman et al.,

2022; Trauger & Tewari, 2024; Yang et al., 2026c) apply to **Empirical Risk Minimization** (ERM), which finds parameters minimizing training error under some constraint on the hypothesis class. We contrast it with the **Guess & Check algorithm** (G&C, Chiang et al., 2023; Buzaglo et al., 2024b), which samples parameter settings from the parameter space until a setting interpolating the training set is found. Guess & Check expands over ERM by modeling the inductive bias of SGD-like optimizers towards solutions occupying large volumes in parameter space (Chiang et al., 2023; Buzaglo et al., 2024a). This is an important difference over bounds based on ERM (Edelman et al., 2022; Yang et al., 2026c; Trauger & Tewari, 2024), as it (i) allows us to explicitly take into account the effect of both the hypothesis class and the ground-truth function, (ii) while bounds depend more mildly on the hypothesis model's size than capacity-based bounds. This enables stronger results when the target function can be represented by small models, and we find it to be borne out empirically in experiments. G&C is closely related to a Bayesian interpretation of learning, as it samples from a posterior over the parameter space conditional on the training set (e.g. Haussler et al., 1994). We throughout work in the realizable regime, where the training data can be exactly fitted by an element of the hypothesis class. Here, ERM and G&C can be defined as follows:

Definition 2.1. *Let a hypothesis class $\mathcal{H} = \mathcal{FPT}(L_s, D_s, H_s)$ (where "$\ldots_s$" stands for "student"), and a training set $(X^{(n)}, Y^{(n)})_{n=1,\ldots,N}$, be given.*

*The **ERM Learner** selects a transformer $T_s \in \mathcal{H}$ such that T_s perfectly fits the training set: $T_s(X^{(n)}) \equiv Y^{(n)} \forall (X^{(n)}, Y^{(n)})_{n=1,\ldots,N}$. The **Guess & Check Learner** returns a sample from the posterior*

$$P\left(T_s \mid T_s(X^{(n)}) \equiv Y^{(n)} \; \forall (X^{(n)}, Y^{(n)})_{n=1,\ldots,N}\right) \tag{1}$$

where $P(T_s)$ is the uniform distribution over $\mathcal{H}$.

Intuitively, G&C adds an inductive bias towards hypotheses represented by a larger volume in parameter space. See Appendix A for discussion beyond the interpolating regime.

Remark 2.1. Throughout this paper, we consider the student–teacher learning framework between two transformers T_s and T_t, with $L_s \geq L_t, D_s \geq D_t$ and $H_s \geq H_t$ for realizability. In particular, we do not require the per-head dimension of the student to dominate that of the teacher (see Appendix E.2).

3 Length Generalization

We first derive length generalization bounds when unbounded data samples are available (Section 3.1); and statistical generalization bounds at a fixed length (Section 4.1). We then derive joint generalization bounds (Section 4.2).

3.1 Deterministic Length Generalization

Our first result states that, if two fixed-precision transformers exactly agree up to a certain length bound C, they must agree everywhere. We in fact obtain two distinct bounds with different rates:

Lemma 3.1 (Deterministic Length Generalization)**.** *Let $r_{\mathbb{F}}$ be a well chosen constant and let $T_s \in \mathcal{FPT}(L_s, D_s, H_s)$ and $T_t \in \mathcal{FPT}(L_t, D_t, H_t)$ be fixed-precision transformers (2.1). Set*

$$C_0 \coloneqq r_{\mathbb{F}}\left[L_s D_s |\mathbb{F}|^{D_{h,s}+1} + L_t D_t |\mathbb{F}|^{D_{h,t}+1}\right] + 1, \qquad C_1 \coloneqq \binom{r_{\mathbb{F}} L_s + |\Sigma|}{|\Sigma|} \tag{2}$$

and let $C \coloneqq \min(C_0, C_1)$. Then, if T_s and T_t agree on every word in $\Sigma^{\leq C}$, they are identical on every word in Σ^*:

$$T_s|_{\leq C} \equiv T_t|_{\leq C} \quad \Longrightarrow \quad T_s \equiv T_t. \tag{3}$$

This statement yields a first length generalization guarantee, in the setting where unbounded data samples are available: In this case, if a learned model T_s exactly fits data representing a target function expressed by T_t up to length C, then we know $T_s \equiv T_t$ even on longer length strings, i.e., length generalization occurs.

See Appendix D for the formal proof; here, we provide an informal discussion to build intuition. Given a teacher transformer T_t implementing the target function and a student T_s from the hypothesis class, we introduce the *difference function* (Figure 2b), which checks for any given word if

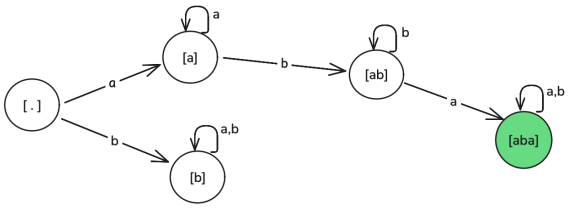
the output vectors of both transformers match at every position. Previously, Li & Cotterell (2025) showed that fixed-precision transformers with a binary classifying output layer are expressively equivalent to **LTL**[**P**], a fragment of linear temporal logic, and further can be represented as a *partially-ordered deterministic finite automaton* (poDFA) (Figure 2a). A poDFA is a DFA where the reachability relation defines a partial order on the state set; intuitively, such automata have no nontrivial loops. Intuitively, the state space describes the dynamic of the transformer's KV-cache; as soon as some activation is added to the KV-cache, it cannot be removed anymore, giving rise to the partially ordered loop-free structure.

Building on this, we express the difference function as a logic formula in **LTL**[**P**], and further as a poDFA $\mathcal{A}_\Delta$ that represents the difference function between the student and the teacher. The student and the teacher transformers produce a different output somewhere on a word $w_1 \dots, w_n$ if and only if $\mathcal{A}_\Delta$, run on the word, ends up in the unique absorbing state $\bot$. As a supplement, we mention in Appendix C, D how transformers exist as ordered Mealy machines (Def. C.5), providing a similar underlying ordered structure on individual transformers, even outside of the learning framework.

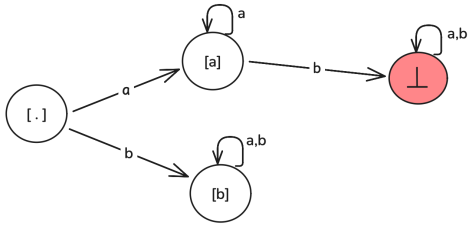
We then upper-bound the *height* of $\mathcal{A}_\Delta$ – i.e., the length of its longest transition chain. The key idea here is that observing behaviour on strings up to roughly this length should be important for length generalization, because strings of this length should be minimally required to be able to observe the whole state space of $\mathcal{A}_\Delta$ – and thus the whole range of behaviour expressed by the student and teacher models. We obtain C_0, C_1 as two upper bounds on the height of $\mathcal{A}_\Delta$ (eq. 2); whichever bound is smaller provides a tighter bound in a given setting, though neither dominates the other globally. Notably, we show some kind of tightness in L and $\mathbb{F}$ of our bounds in Appendix D.5, where we prove that any upper bound on the height must depends at least linearly on $|\mathbb{F}|$ and L.

Now, if T_s and T_t agree up to length $C = \min\{C_0, C_1\}$, $\mathcal{A}_\Delta$ cannot visit a $\bot$ state on any word of this length. But given that C upper-bounds the height of the automaton, the $\bot$ state cannot be reachable even by longer inputs, and $T_s \equiv T_t$ across Σ^*.

The above entails a first length-generalization bound in the limiting case where all data up to some length is available; it is an illustrative first special case of our general result.



(a) Partially-ordered DFA verifying on a word-level, the language $L_3 \coloneqq a\Sigma^* b\Sigma^* a\Sigma^*$.



(b) poDFA of the *difference function* between languages $L_2 \coloneqq a\Sigma^* b\Sigma^*$ and $L_3 \coloneqq a\Sigma^* b\Sigma^* a\Sigma^*$. Here, membership for L_2 and L_3 are evaluated at every position. In particular, one disagreement for some position condemn the run to $\bot$.

Figure 2: Examples of poDFAs: the vertices are the states and the arrows represent the transition function δ.

3.2 Extension to the probabilistic framework

We next ask for a similar length generalization guarantee when T_s and T_t are equal only on *most* strings up to length K. Setting the data distribution $\mathcal{D}_K \coloneqq \mathrm{Unif}(\Sigma^{\leq K})$ and using the bound C from Lemma 3.1, we indeed find a probabilistic extension of the guarantee:

Lemma 3.2 (Probabilistic Length Generalization)**.** *Let $\epsilon > 0$ and consider the training words of length up to K, with*

$$K \geq |\Sigma| \left(C + \log \frac{1}{\epsilon} + \sqrt{\log^2 \frac{1}{\epsilon} + 2C \log \frac{1}{\epsilon}} \right) \sim \mathcal{O}\big(|\Sigma|(C + \log \frac{1}{\epsilon})\big). \tag{4}$$

Then

$$\Pr_{\mathcal{D}_K}[T_s(w) \neq T_t(w)] \leq \epsilon \quad \Longrightarrow \quad \forall M \geq K,\ \Pr_{\mathcal{D}_M}[T_s(w) \neq T_t(w)] \leq 3\epsilon.$$

See Appendix D.4 for the formal proof. Intuitively, this holds for two reasons: (i) the poDFA $\mathcal{A}_\Delta$ ultimately reaches a sink *absorbing* state; (ii) the input distribution $\mathcal{D}_K$ is uniform and so are the letters: this implies that for any non-absorbing state, the probability to leave the state by reading the next letter is at least of $1/|\Sigma|$. Therefore, after a big enough number of steps K, $\mathcal{A}_\Delta$ will be likely to live in an absorbing state. In particular, since the probability of a transition is at least $1/|\Sigma|$, the overall probability of a word to reach an absorbing state after K steps dominates the binomial law $Y \sim \mathrm{Bin}(K, 1/|\Sigma|)$. The lemma follows from the application of a Chernoff bound.

4 Joining Length and Statistical Generalization

4.1 Data complexity

We lift the deterministic statement to a statistical generalization statement. The key here is to combine Theorem 3.1 with statistical in-distribution generalization on the strings up to length K:

Proposition 4.1 (Statistical Generalization for ERM and G&C)**.** *Fix $\epsilon \in (0,1)$ and $\delta \in (0, \frac{1}{5})$. Assume the target function is represented by a teacher $T_t \in \mathcal{FPT}(L_t, D_t, H_t)$, and the learner (G&C or ERM) selects a student T_s from the hypothesis class $\mathcal{FPT}(L_s, D_s, H_s)$. Let S be an i.i.d. sample of N strings from $\mathcal{D}_K$.*

1. *Let T_s be obtained by the G&C learner on S. If*

$$N \geq \frac{1}{\epsilon}\big(B \log|\mathbb{F}| + 3\log(2/\delta)\big), \qquad B \sim \mathcal{O}\big(L_t D_t^2 + D_s D_t(2L_s + 3L_t) + |\Sigma| D_s\big),$$

then, with probability at least $1-\delta$ over S and the draw of T_s,

$$\Pr_{w \sim \mathcal{D}_K}\big[T_s(w) \neq T_t(w)\big] < \epsilon.$$

2. *Let T_s be obtained by ERM on S. If*

$$N \geq \frac{1}{\epsilon}\big(B_{ERM} \log|\mathbb{F}| + \log(1/\delta)\big), \qquad B_{ERM} \sim \mathcal{O}(L_s D_s^2),$$

then, with probability at least $1-\delta$ over S,

$$\Pr_{w \sim \mathcal{D}_K}\big[T_s(w) \neq T_t(w)\big] < \epsilon.$$

Note here that $\mathcal{O}(\cdot)$ absorbs constants depending on $\mathbb{F}$ and ν. The ERM bound has a similar flavor to bounds using capacity-based arguments (Edelman et al., 2022; Trauger & Tewari, 2024; Yang et al., 2026c) in that it scales with parameters count of the hypothesis class. The G&C bound has a similar dependency but scales with dd^* rather than d^2. This provides improved generalization on low-complexity targets, such as larger student networks.

This result extends the work of Rizvi-Martel et al. (2025) with some improvements; we discuss the proof and extensions in Appendix E. For ERM, this is a straightforward application of Appendix B.1 in Buzaglo et al. (2024b). For G&C, the proof strategy (Appendix E) is to show that low-complexity teachers can be embedded in many equivalent ways into the parameter space of a higher-complexity hypothesis class; this means that G&C is intrinsically more likely to find low-complexity interpolators. This insight (Buzaglo et al. (2024b)) drives the appearance of a teacher complexity term, and a milder dependence on student complexity than based on hypothesis class parameter count alone.

4.2 Joint Statistical and Length Generalization

We now combine length and statistical generalization to obtain our main result. The basic intuition is that generalization occurs when (i) the length is sufficiently long so that all representative behavior relevant at long lengths are already visible, (ii) the sample size is sufficient for achieving low test error at this length; combining (i) and (ii) then entails low test error at higher lengths. The result is:

Theorem 4.1 (Statistical and Length Generalization). *Let the transformers $T_s \in \mathcal{FPT}(L_s, D_s, H_s)$ student and $T_t \in \mathcal{FPT}(L_t, D_t, H_t)$ teacher satisfying 2.1, C as in Eq. 2, $\epsilon \in (0,1)$ and $\delta \in (0, \frac{1}{5})$.*

If T_s is given by the Guess & Check procedure, agreeing with T_t on N i.i.d. samples from $\mathcal{D}_K$, with

$$N \geq \frac{1}{\epsilon}\big(B \log|\mathbb{F}| + 3\log(2/\delta)\big) \sim \mathcal{O}\left(\frac{1}{\epsilon}[(L_t D_t^2 + D_s D_t(2L_s + 3L_t)) + 3\log(2/\delta)]\right) \tag{5}$$

and

$$K \geq |\Sigma|\left(C + \ln(1/\epsilon) + \sqrt{\ln^2(1/\epsilon) + 2C\ln(1/\epsilon)}\right) \sim \mathcal{O}\big(|\Sigma|(C + \ln(1/\epsilon)\big), \tag{6}$$

with

$$C \sim \mathcal{O}\left(L_s D_s \exp(D_{h,s}) + L_t D_t \exp(D_{h,t})\right) \quad \textit{or} \quad C \sim \mathcal{O}\left(L_s^{|\Sigma|}\right) \tag{7}$$

then with probability $1-\delta$, we have

$$\forall M \geq K, \qquad \Pr_{\mathcal{D}_M}\left[T_s(w) \neq T_t(w)\right] \leq 3\epsilon. \tag{8}$$

See Appendix F.1 for the formal proof. For ERM, an analogous result ensure from bounding N appropriately, following Proposition 4.1. In addition, the theorem can be adapted to alternative word distributions $\widetilde{D}_K$ sampling enough long words; we give an example in Appendix F.4.

4.3 Tradeoff between Length and Data Complexity

The previous theorem provides conditions on the data complexity N and the training length threshold K, such that low loss can be achieved. Assuming the same learning framework for student T_s, teacher T_t, we now investigate how these quantities trade off, by giving an error bound for any joint choice of N and K:

Theorem 4.2 (Joint Generalization Bound in Sample Size and Length). *Define*

$$a_C(K) := \Pr\left(Bin(K, \frac{1}{|\Sigma|}) \geq C\right), \quad \rho := \frac{|\Sigma|}{|\Sigma|-1} \quad \textit{and} \quad B_\delta := B\log|\mathbb{F}| + 3\log(2/\delta), \tag{9}$$

where C is the bound given in Lemma 3.1 and B is the sample complexity term from Proposition 4.1.

Now consider that the student is trained on N random samples from $\mathcal{D}_K$ and that K is big enough so that $a_C(K) \neq 0$. Then with probability $1-\delta$, for any length $M \geq K$, the error rate between the student and the teacher is bounded as

$$\epsilon := \Pr_{\mathcal{D}_M}\left[T_s(w) \neq T_t(w)\right] \leq \frac{\rho B_\delta}{N \cdot a_C(K)}. \tag{10}$$

See Appendix F.2 for the proof. This value $a_C(K)$ lower-bounds the probability of reaching an absorbing state by time K, and $a_C(K) = 0$ when $K < C$. This theorem holds for either of the Guess & Check or the ERM learner, setting B accordingly following Proposition 4.1. Moreover, under an appropriate definition of $a_C(K)$, the theorem even remains valid if the input strings are generated from appropriate nonuniform distributions (Appendix G.2). We further complement the upper bound (10) with a lower bound: there are families of target functions and input distributions such that the expected test error on long inputs is lower-bounded by $\frac{B}{8Na_C(K)}$ for an appropriate choice of B (Appendix, Theorem G.3).

Takeaway: Importance of Long Training Samples for Transformers. This implies the following: First, when $K < C$, there is no expectation for length generalization. Second, when K just slightly exceeds C, length generalization can be guaranteed, but only at an extraordinary sample size on the order of $\gtrapprox |\Sigma|^C$. Namely, $a_C(C) = s^{-C}$, leading to a sample size requirement $N_{req}(C) = N_0 s^C$ where $N_0 = \frac{\rho B_\delta}{\epsilon}$. Intuitively, this happens because only a small fraction of the samples will already have reached an absorbing state; most samples will thus not be informative about the target function's behavior on long strings.

Third, increasing K by a constant can reduce the required sample size by a multiplicative factor. Specifically, increasing length from C to sC reduces the required sample size by a factor of at least $s^C/2$; small increases to K already massively decrease the sample size requirement // the increase of K by a linear rate (in C) permits a saving in N of an exponential rate (in C).

Given a fixed training-token budget $Budget = NK$, the above considerations allow us to determine an optimal allocation to N and K; we show in Appendix F.3 that, for $C \geq 2$, the following closed form achieves an error bound within a factor $1 + \mathcal{O}(\sqrt{\log C/C})$ of the optimum:

$$\widehat{K} := \left\lceil |\Sigma| \left(C + \log C + \sqrt{\log^2 C + 2C \log C} \right) \right\rceil .$$

Intuitively, this means that close-to-optimal budget allocation can be achieved at around $K \sim |\Sigma|C$, exponentially more efficient that at $K \sim C$.

Our result reveals that there are several regimes of generalization. When K is too small, there is in general no hope of length generalization no matter how large N is. When K has just crossed the threshold given by the longest chain, one may need a very large N, because only few strings will have reached the final state. In this regime, increasing K further is much more efficient than increasing N: increasing K by an additive constant can cut down N by a multiplicative factor.

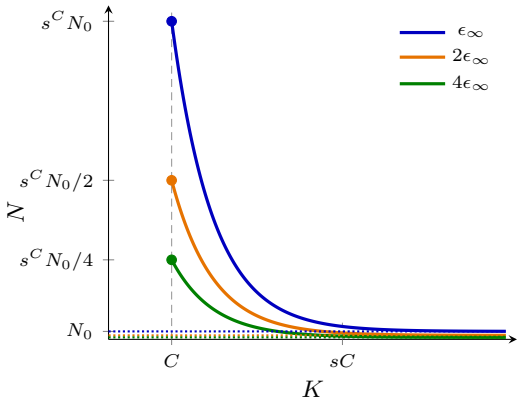


Figure 3: Schematic error-budget limit lines with linear axes. At $K = C$, every budget can be respected with $N \sim \mathcal{O}(s^C)$ training samples. By length $K = sC$, at least half of the words have reached their terminal absorbing states, so each curve is within a factor 2 of its floor; further length gains saturate.

4.4 Extension with Local Attention

Local attention has emerged as an effective and popular addition to transformers (e.g. Beltagy et al., 2020). It is also relevant because positional information often specifically introduces a locality bias favoring attention to recent position (Press et al., 2021; Su et al., 2024). Local attention thus simultaneously serves as a simplified account of the role of positional information (similar to positional functions in Limit Transformers, Huang et al. (2025a); Izzo et al. (2025)). In each layer, we add a certain number of local attention heads at some window size W; for these heads, context positions at distance $> W$ are masked. Both W and the number of local heads are hyperparameters. We refer to Li & Cotterell (2026) for the full formalization in the FPT context.

Formally, we can model an FPT with local attention in terms of a more complex automaton that arises as a *cascade product* of a poDFA with a simple automaton that maintains the last $L \cdot W$ symbols. That is, we can model the difference automaton $\mathcal{A}$ in terms of the cascade product of a poDFA representing the output difference between the two transformers with a simple automaton that maintains the last $L \cdot W$ symbols. More details in Appendix G.1.

The statistical generalization guarantee (Proposition 4.1) is unaffected, because the number of parameters does not change when making some heads local. The length generalization guarantee changes. First, the construction we provide is only valid for the bound C_0 in Eq. 2. Second this threshold C_0 receives a factor of WL. Third, the bounding binomial probability to go forward in the underlying poDFA is also divided by WL. Thus, as long as W is fixed, we have the same qualitative predictions as above: model size impacts data and length complexity; and sample size and length trade off.

5 Experiments

We introduce three families of tasks, and study (i) the impact of teacher and student complexity on length and data complexity, and (ii) the predicted interaction between length and sample size.

Tasks. We defined three task families, each along a complexity scale indexed by $k \in \mathbb{N}$, as follows (i) **Alternating languages L_k:** Detecting whether $\geq k$ distinct blocks of a's and b's have occurred (Yang et al., 2025a) – that is, checking membership in the languages $L_2 = a^+b^+\Sigma^*$; $L_3 = a^+b^+a^+\Sigma^*$, etc. There is a teacher family with number of layers $L_k = \Theta(k)$, embedding dimension $D_k = O(1)$, head count $H_k = O(1)$ and fixed alphabet size $|\Sigma|_k = O(1)$. Specifically, the letter sampling is not uniform, with probability to alternate block set to 0.05. This still fits within our uniform theoretical framework, as explained in Appendix H.1.1. (ii) **Alphabet languages A_k:** Detecting whether all symbols from a given alphabet $\Sigma_k = [k]$ have occurred. There is a teacher family with $L_k = O(1)$, $D_k = O(k)$, $H_k = O(k)$, $|\Sigma|_k = O(k)$. (iii) **Induction Head tasks IH_k:** Outputting which next tokens can follow the current token under the context's bigram distribution over the alphabet $\Sigma_k = [k]$ (cf. Edelman et al., 2024); that is, given the current token α output a vector $\in \mathbb{F}^{|\Sigma|}$ indicating for which symbols β the bigram $\alpha\beta$ has contextually occurred. There is a teacher family with $L_k = O(1)$, $D_k = O(k)$, $H_k = O(1)$, $|\Sigma|_k = O(k)$, with local attention. See Appendix H for teachers constructions. Taken together, these tasks allow us to test the theoretical prediction that the teacher size (impacted by L, H, D, $|\Sigma|$) impacts both length and data complexity.

Setup. To test our predictions against ordinary gradient-based training, we train FP32 transformers with Adam Kingma & Ba (2014) and binary cross-entropy over 10 seeds. We use binary classification because exact logit-equality in FP32 is not realistic. The difference function between the student and the teacher then checks for logit signs rather than exact values (Appendix H.2). In addition, we retain only runs with perfect training accuracy—that is, runs that correctly classify every supervised target on every training word. This enforces the conditioning of Guess & Check. Trained architectural sizes are fixed and common among a family, unless explicitly varied. When sweeping N, we adjust K across tasks to match the fraction of words exhibiting the required blocks, letters, or bigrams, making examples approximately comparable in informativeness (Appendix H.2). Sweeps of K instead compare tasks at common lengths.

Effect of Transformers Size on Data and Length Complexity. Across the three families, increasing teacher size impacts the sample size N and training length K at which generalization becomes successful. (Figure 4). the predicted complexity costs. This is in line with our theoretical predictions, in particular, the G&C-based bounds. Interestingly, the student size had relatively small impact on sample size and length (Appendix H.3.2). Thus, G&C-based bounds are successful in capturing the role of the student size but still may be overly pessimistic about it.

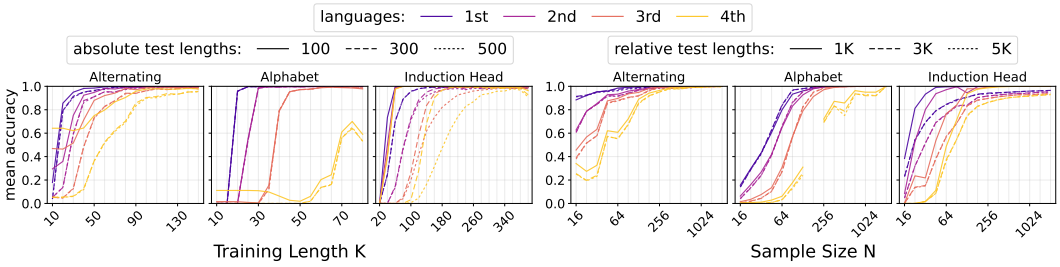


Figure 4: **Left:** Impact of teacher's size on length complexity. Larger teachers require training samples to be longer. The abscissa scale is linear, as suggested by theory. **Right:** Impact of teacher's size on data complexity. Larger teachers require more training samples for similar length generalization performance. The abscissa scale is exponential, as suggested by theory. On both sides, the plotted tasks are respectively the Alternating L_4, L_5, L_6, L_7, the Alphabet $A_{10}, A_{12}, A_{15}, A_{20}$ and the Induction Head IH_5, IH_6, IH_7, IH_8.

Tradeoff between Sample Size and Length. Longer training words can substantially reduce the sample size needed for generalization (Figure 5). In particular, we find the non-linear interaction between the two quantities, qualitatively in line with the predictions from Figure 3. This backs the idea that an efficient token-budget allocation should ensure that training samples are long enough. See Appendix H.3.3 for further results showing the same pattern.

6 Discussion

Our result goes beyond established theory in two ways: First, it goes beyond **statistical generalization** bounds for transformers by guaranteeing length generalization. Prior learning results for trans-

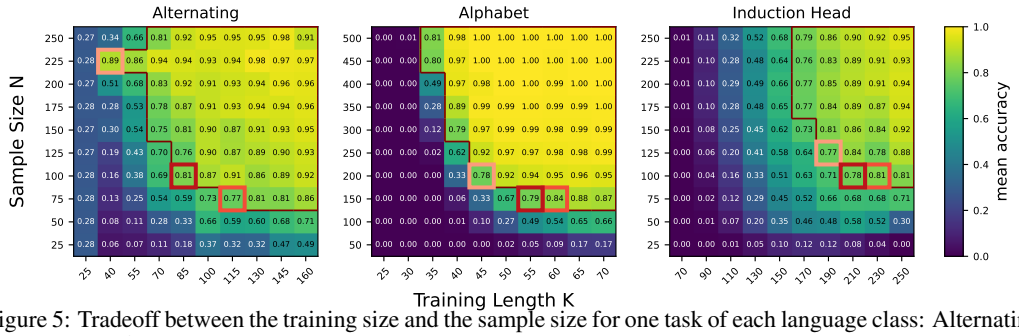


Figure 5: Tradeoff between the training size and the sample size for one task of each language class: Alternating L_6, Alphabet A_{15} and Induction Head IH_7. Evaluation at length 5 times the training length. Contouring around cells with accuracy ≥ 0.75; under this condition, the 3 cells with the smallest token-budget are framed in red.

formers characterize generalization error and sample size when the test and train samples are at the same sequence length (Edelman et al., 2022; Trauger & Tewari, 2024; Li et al., 2026; Truong, 2024; Yang et al., 2026c; Li et al., 2025; Rizvi-Martel et al., 2026; Lintilhac & Shaikh, 2026; Mousavi-Hosseini et al., 2025; Hanneke et al., 2026; Joshi et al., 2025). Our results instead consider the more general case of length generalization, in the setting of fixed-precision transformers. We note that our main result ingests the statistical generalization bound essentially as a black box; hence, our results might analogously also be formalized on the basis of other statistical generalization bounds.

Second, our results go beyond prior work on **length generalization** by giving explicit and joint rates for sample size and length. So far, the most generally predictive account of length generalization has been the RASP-L Conjecture (Zhou et al., 2024) and its formalization via C-RASP and Limit Transformers (Huang et al., 2025a). This theory has been highly predictive of which tasks empirically show length generalization (Yang et al., 2026b), but provides no quantitative predictions. Notably, C-RASP is more powerful than FPT; e.g. it can model the MAJORITY language. However, C-RASP is a too loose target for quantitive learning bounds: Yang et al. (2026a) show that, beyond the shallow-depth case (Izzo et al., 2025; Chen et al., 2025), any length threshold would need to grow uncomputably with model complexity. The positive fragment C-RASP$^+$ has expressivity equivalent to fixed-precision transformers and was studied in Yang et al. (2026a); Zetzsche et al. (2026). We compare with these results in detail in Appendix A. Importantly, our work innovates over these prior results by providing joint generalization guarantees in both sample size and length.

Like capacity-based generalization bounds (Edelman et al., 2022; Trauger & Tewari, 2024; Yang et al., 2026c), our work treats an idealized learning model that abstracts away from the computational process of finding a model fitting the training data. Understanding optimization dynamics for transformers has been difficult. Progress has been made for specific task families and restricted models, such as single layers (e.g. Yang et al., 2025b; Mwigo & Dasgupta, 2026; Huang et al., 2025b; Wang et al., 2024) (including one-layer models performing chain-of-thought, Huang et al. (2025b)) or specific task families (e.g. Nichani et al., 2024). We view the study of G&C as an (imperfect) proxy for part of the implicit inductive bias of SGD-like learning; understanding of optimization dynamics for a general multi-layer model and a general hypothesis class will require new methods.

The main limitation of our results is that they predict asymptotic rates but not yet concrete realistic numerical bounds for specific tasks and models. The primary barrier here is that the bounds involve large constants involving the fixed-precision set $\mathbb{F}$, which is large at full-precision training. It is an interesting question whether realistic numerical bounds can be obtained when training with strong quantization. Further limitations are that fixed-precision transformers are less expressive than log-precision or infinite-precision transformers, we did not treat positional encodings, and studied G&C and ERM as opposed to optimization dynamics.

7 Conclusion

We have provided generalization bounds for multi-layer fixed-precision transformers that account for the roles of both the number and the length of training samples. We provided a joint treatment of both dimensions, showing that, under a given token budget, increasing length can sometimes be exponentially more efficient than increasing sample size.

REFERENCES

Yotam Alexander, Yonatan Slutzky, Yuval Ran-Milo, and Nadav Cohen. Do neural networks need gradient descent to generalize? a theoretical study. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 143774–143840. Curran Associates, Inc., 2025. doi: 10.52202/085713-4813. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/d39fb2054215f07d1f90cc80c7a85edd-Paper-Conference.pdf.

Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. *Advances in neural information processing systems*, 32, 2019.

Cem Anil, Yuhuai Wu, Anders Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur. Exploring length generalization in large language models. *Advances in Neural Information Processing Systems*, 35:38546–38556, 2022.

Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 322–332. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/arora19a.html.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer Normalization. *arXiv preprint arXiv:1607.06450*, 2016. URL https://arxiv.org/abs/1607.06450.

Peter L Bartlett, Andrea Montanari, and Alexander Rakhlin. Deep learning: a statistical viewpoint. *Acta numerica*, 30:87–201, 2021.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. On the ability and limitations of transformers to recognize formal languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7096–7116, 2020.

Janusz A. Brzozowski and Faith E. Fich. Languages of R-trivial Monoids*. *Journal of Computer and System Sciences*, 20:32–49, 1980. doi: 10.1016/0022-0000(80)90003-3. URL https://www.sciencedirect.com/science/article/pii/0022000080900033.

Gon Buzaglo, Itamar Harel, Mor Shpigel Nacson, Alon Brutzkus, Nathan Srebro, and Daniel Soudry. How Uniform Random Weights Induce Non-uniform Bias: Typical Interpolating Neural Networks Generalize with Narrow Teachers. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 5035–5081. PMLR, 21–27 Jul 2024a. URL https://proceedings.mlr.press/v235/buzaglo24a.html.

Gon Buzaglo, Itamar Harel, et al. How Uniform Random Weights Induce Non-uniform Bias: Typical Interpolating Neural Networks Generalize with Narrow Teachers. In *International Conference on Machine Learning (ICML)*, 41, 2024b. URL https://arxiv.org/abs/2402.06323.

Thomas Chen, Tengyu Ma, and Zhiyuan Li. Non-asymptotic length generalization. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 8621–8682. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/chen25ar.html.

Ping-yeh Chiang, Renkun Ni, David Yu Miller, Arpit Bansal, Jonas Geiping, Micah Goldblum, and Tom Goldstein. Loss landscapes are all you need: Neural network generalization can be explained without the implicit bias of gradient descent. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=QC10RmRbZy9.

Benjamin L Edelman, Surbhi Goel, Sham Kakade, and Cyril Zhang. Inductive Biases and Variable Creation in Self-Attention Mechanisms. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 5793–5831. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/edelman22a.html.

Ezra Edelman, Nikolaos Tsilivis, Benjamin L. Edelman, Eran Malach, and Surbhi Goel. The Evolution of Statistical Induction Heads: In-Context Learning Markov Chains. In *38th Conference on Neural Information Processing Systems (NeurIPS 2024)*, volume 37, 2024. URL https://openreview.net/forum?id=qaRT6QTIqJ.

Steve Hanneke, Idan Mehalel, and Shay Moran. Sample complexity of autoregressive reasoning: Chain-of-thought vs. end-to-end, 2026. URL https://arxiv.org/abs/2604.12013.

David Haussler, Michael Kearns, and Robert E Schapire. Bounds on the sample complexity of bayesian learning using information theory and the vc dimension. *Machine learning*, 14(1):83–113, 1994.

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 *International Conference on Learning Representations (ICLR 2025)*, 2025a. URL https://arxiv.org/abs/2410.02140.

Xinting Huang, Aleksandra Bakalova, Satwik Bhattamishra, William Merrill, and Michael Hahn. Discovering interpretable algorithms by decompiling transformers to RASP. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=QnWjcTk14G.

Yu Huang, Zixin Wen, Aarti Singh, Yuejie Chi, and Yuxin Chen. Transformers provably learn chain-of-thought reasoning with length generalization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=aE0bCvXXBt.

Zachary Izzo, Eshaan Nichani, and Jason D. Lee. Quantitative Bounds for Length Generalization in Transformers, 2025. URL https://arxiv.org/abs/2510.27015.

Selim Jerad, Anej Svete, Jiaoda Li, and Ryan Cotterell. Disentangling the expressivity of rope. *arXiv preprint arXiv:2608.11909*, 2026.

Mayank Jobanputra, Yana Veitsman, Yash Sarrof, Aleksandra Bakalova, Vera Demberg, Ellie Pavlick, and Michael Hahn. Born a transformer–always a transformer? on the effect of pretraining on architectural abilities. *Advances in Neural Information Processing Systems*, 38:95463–95506, 2026.

Nirmit Joshi, Gal Vardi, Adam Block, Surbhi Goel, Zhiyuan Li, Theodor Misiakiewicz, and Nathan Srebro. A theory of learning with autoregressive chain of thought, 2025. URL https://arxiv.org/abs/2503.07932.

R. Kaas and J.M. Buhrman. Mean, median and mode in binomial distributions. *Statistica Neerlandica*, 34(1):13–18, March 1980. doi: 10.1111/j.1467-9574.1980.tb00681.x. URL https://ideas.repec.org/a/bla/stanee/v34y1980i1p13-18.html.

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. The impact of positional encoding on length generalization in transformers. *Advances in Neural Information Processing Systems*, 36:24892–24928, 2023.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Jiaoda Li and Ryan Cotterell. Characterizing the Expressivity of Fixed-Precision Transformer Language Models. In *39th Conference on Neural Information Processing Systems (NeurIPS 2025)*, volume 39, 2025. URL https://arxiv.org/pdf/2505.23623.

Jiaoda Li and Ryan Cotterell. Characterizing the Expressivity of Local Attention in Transformers. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 37485–37507, 2026. doi: 10.18653/v1/2026.acl-long.1739. URL https://aclanthology.org/2026.acl-long.1739/.

Yawen Li, Tao Hu, Zhouhui Lian, Wan Tian, Yijie Peng, Huiming Zhang, and Zhongyi Li. Sharper generalization bounds for transformer, 2026. URL https://arxiv.org/abs/2603.21541.

Zhihao Li, Xue Jiang, Liyuan Liu, Xuelin Zhang, Hong Chen, and Feng Zheng. On the generalization ability of next-token-prediction pretraining. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 34943–34975. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/li25ao.html.

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*, 2024. URL https://openreview.net/forum?id=3EWTEy9MTM.

Paul Lintilhac and Sair Shaikh. A sharper picture of generalization in transformers, 2026. URL https://arxiv.org/abs/2605.20988.

Charles London and Varun Kanade. Pause tokens strictly increase the expressivity of constant-depth transformers. *Advances in Neural Information Processing Systems*, 38:89765–89795, 2025.

Tomáš Masopust and Michaël Thomazo. On the complexity of k-piecewise testability and the depth of automata. In *International Conference on Developments in Language Theory*, pp. 364–376. Springer, 2015.

Alireza Mousavi-Hosseini, Clayton Sanford, Denny Wu, and Murat A. Erdogdu. When do transformers outperform feedforward and recurrent networks? a statistical perspective, 2025. URL https://arxiv.org/abs/2503.11272.

Brian Mwigo and Anirban Dasgupta. Generalization bound for a shallow transformer trained using gradient descent. *Transactions on Machine Learning Research*, 2026.

John Myhill. Finite Automata and the Representation of Events. Technical Report TR-57-624, Wright Air Development Center, Wright-Patterson Air Force Base, Ohio, 1957.

Vaishnavh Nagarajan and J. Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning, 2021. URL https://arxiv.org/abs/1902.04742.

Anil Nerode. Linear Automaton Transformations. *Proceedings of the American Mathematical Society*, 9(4):541–544, 1958. doi: 10.1090/S0002-9939-1958-0135681-9.

Eshaan Nichani, Alex Damian, and Jason D. Lee. How transformers learn causal structure with gradient descent. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 38018–38070. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/nichani24a.html.

Ofir Press, Noah A Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*, 2021.

Michael Rizvi-Martel, Satwik Bhattamishra, Guillaume Rabusseau, and Michael Hahn. From expressivity to sample complexity: Narrow teachers for transformers via c-rasp. In *Conference on Neural Information Processing Systems (NeurIPS) Workshop: What Can('t) Transformers Do?*, volume 39, 2025.

Michael Rizvi-Martel, Satwik Bhattamishra, Guillaume Rabusseau, and Michael Hahn. From Expressivity to Sample Complexity: Narrow Teachers for Transformers via C-RASP, 2026. URL https://arxiv.org/abs/2607.11760.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Jacob Trauger and Ambuj Tewari. Sequence Length Independent Norm-Based Generalization Bounds for Transformers. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li (eds.), *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 1405–1413. PMLR, 2024. URL https://proceedings.mlr.press/v238/trauger24a.html.

Lan V Truong. On rank-dependent generalisation error bounds for transformers. *arXiv preprint arXiv:2410.11500*, 2024.

Zixuan Wang, Stanley Wei, Daniel Hsu, and Jason D. Lee. Transformers Provably Learn Sparse Token Selection While Fully-Connected Nets Cannot. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 51854–51912. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/wang24ca.html.

Andy Yang and David Chiang. Counting like transformers: Compiling temporal counting logic into softmax transformers. *arXiv preprint arXiv:2404.04393*, 2024.

Andy Yang, Michaël Cadilhac, and David Chiang. Knee-Deep in C-RASP: A Transformer Depth Hierarchy. In *39th Conference on Neural Information Processing Systems (NeurIPS 2025)*, volume 38, 2025a. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/354a6e7be4f1a35127746b8147a31bfe-Abstract-Conference.html.

Andy Yang, Pascal Bergsträßer, Georg Zetzsche, David Chiang, and Anthony Widjaja Lin. Length Generalization Bounds for Transformers. In *Forty-third International Conference on Machine Learning*, 2026a. URL https://openreview.net/forum?id=XNXU8BJRT7.

Andy Yang, Blerta Veseli, Corentin Barloy, Michaël Cadilhac, Andreas Krebs, Charles Paperman, Howard Straubing, and Michael Hahn. Algebraic decomposition theory for transformer length generalization, 2026b. URL https://arxiv.org/abs/2608.13433.

Chenxiao Yang, Nathan Srebro, and Zhiyuan Li. Tight Sample Complexity of Transformers. In Steve Hanneke and Tor Lattimore (eds.), *Proceedings of Thirty Ninth Conference on Learning Theory*, volume 336 of *Proceedings of Machine Learning Research*, pp. 6887–6923. PMLR, 29 Jun–03 Jul 2026c. URL https://proceedings.mlr.press/v336/yang26a.html.

Tong Yang, Yu Huang, Yingbin Liang, and Yuejie Chi. Multi-head Transformers Provably Learn Symbolic Multi-step Reasoning via Gradient Descent. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 116194–116285. Curran Associates, Inc., 2025b. doi: 10.52202/085713-3875. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/a8f123480ede36612314d013aba6340c-Paper-Conference.pdf.

Georg Zetzsche, Hongjian Jiang, Andy Yang, Pascal Bergsträßer, Marco Sälzer, David Chiang, and Anthony W. Lin. Length generalization for transformers via compression, 2026. URL https://arxiv.org/abs/2609.08851.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization, 2017. URL https://arxiv.org/abs/1611.03530.

Hattie Zhou, Arwen Bradley, Etai Littwin, Noam Razin, Omid Saremi, Joshua Susskind, Samy Bengio, and Preetum Nakkiran. What algorithms can transformers learn? a study in length generalization. In *International Conference on Learning Representations*, volume 2024, pp. 15898–15926, 2024.

CONTENTS

1 Introduction **1**

2 Background **2**
2.1 Transformers . . . 2
2.2 Learning framework . . . 2

3 Length Generalization **3**
3.1 Deterministic Length Generalization . . . 3
3.2 Extension to the probabilistic framework . . . 4

4 Joining Length and Statistical Generalization **5**
4.1 Data complexity . . . 5
4.2 Joint Statistical and Length Generalization . . . 5
4.3 Tradeoff between Length and Data Complexity . . . 6
4.4 Extension with Local Attention . . . 7

5 Experiments **7**

6 Discussion **8**

7 Conclusion **9**

A FAQ **16**

B Conventions and Background **19**
B.1 Fixed-precision arithmetic . . . 19
B.2 Transformers . . . 20

C Logic **21**
C.1 Past temporal logic . . . 21
C.2 Past two-variable first-order logic . . . 22
C.3 Transformers as logic formulas . . . 23
C.4 Partially ordered automata . . . 24
C.5 Past subformulas and automaton height . . . 26

D Length Generalization **28**
D.1 Bounding the height with past subformulas . . . 29
D.2 Bounding the height with temporal profiles . . . 31
D.3 Deterministic length generalization . . . 32
D.4 Probabilistic length generalization . . . 33
D.5 Dependence of Chain Length on L and $\mathbb{F}$ . . . 34

D.5.1 Necessity of precision dependence 34
D.5.2 Necessity of linear depth dependence 35

E Data Complexity Bound **36**
E.1 Main statement and proof sketch 36
E.2 Smaller student head dimension 37
E.3 Per-position outputs 37
E.4 Comparison to ERM 38

F Main theorem and Tradeoff **39**
F.1 Main theorem 39
F.2 Multiplicative upper bound 39
F.3 Tradeoff between Length and Data Complexity 41
F.4 Adaptive main theorem: uniform training length 42

G Extensions **44**
G.1 Local Attention 44
G.2 Generalization with Nonuniform Distribution over Vocabulary 46
G.3 Lower Bound on Tradeoff 47

H Experiments **50**
H.1 Task families 50
H.1.1 L_k: Alternating languages 50
H.1.2 A_k: Alphabet languages 50
H.1.3 IH_k: Induction Head tasks 51
H.2 Protocol and evaluation 51
H.3 Results 53
H.3.1 Impact of the teacher's size 53
H.3.2 Impact of the student's size 55
H.3.3 Tradeoff between training length and data complexity 57

I AI Usage **66**

A FAQ

1. *How do the results relate to Yang et al. (2026a); Zetzsche et al. (2026)?*

   Yang et al. (2026a) give two core results: (i) an impossibility result for computable length generalization bounds for C-RASP, and (ii) a deterministic length generalization bound analogous to our Lemma 3.1 for C-RASP$^+$.

   The first result indicates that C-RASP is too powerful a target for obtaining quantitative learning results for transformers: even though C-RASP has been successful in predicting generalization behaviour experimentally (Huang et al., 2025a; Yang et al., 2026b), there are C-RASP-expressible tasks where training data would need to include strings with astronomically large lengths. Yang et al. (2026a); Zetzsche et al. (2026) further show an

analogue of our Lemma 3.1 for C-RASP$^+$, the positive fragment of C-RASP, giving a computable bound in the size of the defining C-RASP formula. Most recently, Zetzsche et al. (2026) have further studied the role of sample size. They show exponential sample size bounds for C-RASP$^+$ and another fragment, called C-RASP$_1$.

C-RASP$^+$ has the same expressivity as fixed-precision transformers, though there is no direct 1-to-1 link between the transformer size parameters (L, D, H) and the size of defining C-RASP$^+$ formulas. As a consequence, length and sample complexities that we obtain may not be directly comparable to bounds obtained for C-RASP$^+$ in Yang et al. (2026a); Zetzsche et al. (2026).

An important difference of our results, besides directly applying to a transformer model rather than the C-RASP$^+$ logic, is that we obtain joint bounds allowing us to understand the tradeoff between sample length and sample size.

2. *How do the experiments differ from the theoretical learning model? What are the implications of these differences?*

   The point of our experiments is to test theoretical predictions (obtained for idealized learners G&C or ERM on fixed-precision transformers) in a realistic training setup using gradient-based optimization (Adam) on high-precision (FP32) transformers. Whereas the theoretical setup asks for exact match of the output vectors, the experiments instead train for per-dimension binary cross-entropy. In experiments, we select only runs with perfect training accuracy, enforcing interpolation, as assumed in theory.

3. *Why study fixed-precision transformers? Why not study C-RASP / Limit Transformers?*

   It is a well-established empirical finding that transformers show strong length generalization on the class of languages expressed by fixed-precision transformers (Li & Cotterell, 2025; Yang et al., 2026b). They are thus a natural target for joint treatment of sample and length complexity.

   Fixed-precision transformers define a strict subset of C-RASP (Yang & Chiang, 2024) and Limit Transformers (Huang et al., 2025a; Izzo et al., 2025). On the one hand, there is empirical evidence that transformers also show length generalization on tasks that are expressible in C-RASP but not by fixed-precision transformers, such as MAJORITY or certain regular languages such as bounded-depth Dyck languages (Yang et al., 2026b), which require a transformer's numerical precision to grow with the input length. On the other hand, as discussed above, Yang et al. (2026a) show that C-RASP cannot admit computable length generalization bounds, i.e., any threshold C satisfying (3) for all of C-RASP must grow faster in the size of the C-RASP formula (or Limit Transformer) than any computable function. This indicates that C-RASP is too broad as a target for quantitative length generalization bounds (Yang et al., 2026a; Zetzsche et al., 2026). It is an interesting question for future work whether tractable generalization bounds exist for some natural class between fixed-precision transformers and C-RASP.

4. *How do the results relate to Izzo et al. (2025); Chen et al. (2025)?*

   Izzo et al. (2025); Chen et al. (2025) provide quantitative length generalization bounds for C-RASP or Limit Transformers fragments with only one or two layers. On the one hand, fixed-precision transformers form a strict subset of C-RASP/Limit Transformers; on the other hand, our results apply to the multi-layer setup, where computable quantitative bounds for C-RASP are impossible (see Items 1 and 3).

5. *The paper studies ERM and G&C. Why not study optimizers like SGD or Adam?*

   Our work follows prior work (e.g. Edelman et al., 2022; Yang et al., 2026a; Chen et al., 2025; Izzo et al., 2025; Trauger & Tewari, 2024) in studying idealized learners abstracting away from gradient dynamics. ERM in particular is a highly popular standard approach for deriving learning bounds. Guess-and-check models the inductive bias of realistic training towards solutions occupying high volume in parameter space Buzaglo et al. (2024b), unlike ERM, but neither G&C nor ERM could be considered full models of learning. For instance, for matrix factorization, Alexander et al. (2025) show that SGD can in some cases of matrix factorization generalize even better than G&C. Understanding gradient dynamics for general mutli-layer transformers is extremely difficult. Developing quantitative generalization bounds for gradient-based training is a major task for future work; the 1-layer setup might be a tractable starting point.

6. *Could the framework be extended to handle positional encodings?*

   Prior bounds for statistical generalization (not length generalization) of transformers use fixed absolute positional encodings (Edelman et al., 2022; Yang et al., 2026c; Lintilhac & Shaikh, 2026). That would also be possible to include in our setup, though one would need to understand their effects on the automaton structure. Learnable absolute positional encodings are harder to account for as they lead to rapid growth of the parameter count with the input length (as acknowledged by Edelman et al. (2022)), and are inherently undefined outside of the training length range. Huang et al. (2025a) impose translation invariance constraints to constrain the hypothesis space in their analysis.

   Relative positional encodings can be easier to account for, as they do not involve an unbounded number of parameters. Yang et al. (2025a) show that ALiBi (Press et al., 2021) can be understood (in the context of C-RASP) as the addition of local predicates; we expect the effect on fixed-precision transformers to be very similar to the effect provided by local attention. We opt to analyze local attention, as it is nowadays popular in LLM design and has been rigorously formalized in the fixed-precision setup by Li & Cotterell (2026). RoPE (Su et al., 2024) is harder to capture theoretically in the context of length generalization: while it can be formalized in terms of modular predicates (Yang et al., 2025a) (which admit exact automata-theoretic treatment in an appropriate extension of our poDFA framework, with appropriate modifications to the length complexity term), this does not characterize conventional RoPE implementation, which rather behave like fixed-length backwards-lookup operators (Jerad et al., 2026), and may in fact impair empirical length generalization (Jerad et al., 2026).

7. *Can the theory predict specific numerical bounds?*

   Our aim is to predict qualitative rates and the shape of the tradeoff between sample size and input length. Predicting numerical bounds faces the challenge that constants depending on the precision set $\mathbb{F}$ will be extremely large at full-precision training. Constants involving it are in general needed (Appendix D.5). It is a highly interesting (and challenging) problem for future work to make bounds numerically tighter.

8. *Are the teacher constructions for the three task families minimal?*

   For the Alternation Languages, Yang et al. (2025a) show that transformer depth for L_k must be $\geq k$ under a more expressive formalization of transformers (fixed-precision residual stream, unbounded-precision attention). Thus, our teacher family with $L_k = \Theta(k), D_k = O(1), H_k = O(1)$ is optimal up to rates.

   For the Alphabet Languages, we found models hard to train when head count was $< k$, though we do not have a proof that FPT requires $\Omega(k)$ heads. However, a simple information-theoretic argument shows that $D_k \cdot |\mathbb{F}| = \Omega(\log k)$ is unavoidable; hence, the teachers do need to grow with k. For the Induction Head Languages, the same argument applies as $k \to \infty$.

9. *Why are there missing points in some experimental plots?*

   As detailed in Appendix H.2, across all configurations, each run over 10 random seeds, we strictly retain only the runs that achieve 100% training accuracy, consistent with the Guess & Check conditioning assumption. Consequently, certain challenging configurations show missing values when none of the runs converge:

   - Configurations where the student architecture is too small to express the target function (Appendix H.3.2).
   - Configurations where the student's capacity seems tight with respect to its teacher. For example in Figure 4, the Alphabet language A_{20} struggles to be learned as the student has exactly 20 attention heads (H.3.1). As noted in Appendix H.1 and H.3.3, learning is difficult when capacity exactly matches task complexity with no redundancy. We hence surmise for this case that the sample size $N = 181 \simeq 128\sqrt{2}$, with sparse enough data, provide bad conditions so that sampling variance can steer the optimization away from a perfect training accuracy solution across all seeds.

10. *What is strict causal masking? Why assume it?*

    Strict causal masking means that attention heads cannot attend to the query position, but only to strictly preceding positions. Strict causal masking increases expressivity compared to non-strict causal masking (Li & Cotterell, 2025). We believe that our techniques and

main results would equally apply in the non-strict-masking setup, though then applying to a less expressive model and correspondingly a smaller class of functions.

11. *The theoretical bounds include effects of both teacher size (the transformer representing the target function) and the student size (the transformers in the learner's hypothesis class). Experiments show strong effects of the teacher but weak effects for the student. Why?*

    Our theoretical predictions provide worst-case sufficient upper bounds on the test error; practical learning outcomes may be better both because the theoretical analysis may not be tight and because gradient-based learning may generalize better than the theoretical idealizations ERM and G&C. Indeed, it is a common observation that key techniques for proving statistical learning bounds show an overly pessimistic dependency on the student size: overparameterized neural networks often generalize very well, far better than is accounted for by capacity-based bounds that guarantee learning for ERM via uniform convergence (e.g. Nagarajan & Kolter, 2021; Zhang et al., 2017; Bartlett et al., 2021). The G&C analysis as developed by Buzaglo et al. (2024b), that we adopt in Section 4.1, goes some way towards ameliorating this, as its dependence on the student size is substantially milder than for the capacity-based ERM bounds. Some work has provided generalization bounds that scale very mildly in the student size for simpler neural networks (e.g. Allen-Zhu et al., 2019; Arora et al., 2019), though the extension of such results to transformers will likely require new techniques.

12. *The bound C on the height of the difference automata is chosen as the minimum of two bounds (Lemma 3.1). When is which bound stronger? What do we know about tightness?*

    C_0 involves the layer count, hidden dimension, and per-head dimension of both student and teacher; it is exponential in the per-head dimension and linear in the others. C_1 involves only the student layer count and the alphabet size; it can be exponential in either. Importantly, C is the *minimum* both, i.e., our results always apply to the smaller of both bounds. Thus, even if our theorems were stated just for $C := C_0$ or $C := C_1$, they would remain valid; giving two versions strengthens our results.

    The two bounds have strengths in different hyperparameter regimes. In the regime where models are very deep, C_0 is asymptotically smaller, and C remains *linear* in L_s and L_t. In the regime where models are very wide, C_1 is asymptotically smaller, as it does not depend on the hidden dimension. The exception is when the number of heads also increases: then, the per-head dimension $D_{h,s}$ decreases, and C_0 may become more favorable.

    Notably, only two quantities enter both bounds: L and $|\mathbb{F}|$; both bounds additionally require other quantities, but they can trade off which each other; e.g., as discussed above, taking into account D can turn the depth dependence from exponential (in C_1) to linear (in C_0). We show the unavoidability of including the two quantities L and $|\mathbb{F}|$ in Appendix D.5.

B Conventions and Background

We specify the fixed-precision transformer model used throughout the paper, following Li & Cotterell (2025, Section 3 and Appendix B). Our main adaptations are that we retain the number of attention heads explicitly. In addition, we distinguish tokenwise vectorial outputs from their word-level binary satisfaction framework.

B.1 Fixed-precision arithmetic

Fixed-precision set. A *fixed-precision set* or *finite-precision set* $\mathbb{F}$ is a finite ordered set, constructed from a subset of the real numbers $\mathbb{F}_{\text{fin}}$ containing 0 and 1, and extended with the two infinite limits:

$$\mathbb{F} = \mathbb{F}_{\text{fin}} \cup \{-\infty, +\infty\}, \qquad \{0, 1\} \subseteq \mathbb{F}_{\text{fin}} \subset \mathbb{R}, \qquad p := |\mathbb{F}|. \tag{11}$$

The set $\mathbb{F}$ is equipped with deterministic rules for rounding every elementary operation back to $\mathbb{F}$, including overflow and underflow. When we study transformers on a fixed-precision set $\mathbb{F}$, all scalar parameters and intermediate calculation values must belong to $\mathbb{F}$. The precision convention is fixed and independent of the input length.

We use the natural extended-real rules whenever defined, in particular

$$\exp(-\infty) = 0, \qquad \exp(+\infty) = +\infty, \qquad f/(\pm\infty) = 0 \quad (f \in \mathbb{F}_{\text{fin}}).$$

More details about the edge cases stemming from the finite-precision arithmetic can be found in Li & Cotterell (2025).

B.2 TRANSFORMERS

Transformers read words $w = w_1 \cdots w_n \in \Sigma^*$ directly; throughout, $n = |w|$ refers to the length of an input word. Differently to Li & Cotterell (2025), no end-of-sequence token is appended, and transformer computations are performed only at the ordinary positions $1, \ldots, n$.

We use i, j for positions, ℓ for layers, d for the vectorial dimension and h for heads.

Definition B.1 (Fixed-precision transformer)**.** *Fix a depth $L \geq 0$, a width $D \geq 1$, and a number of heads $H \geq 1$ with $H \mid D$; put $D_h = D/H$. A fixed-precision transformer assigns to each $w \in \Sigma^*$ of length n an output*

$$T(w) \in \mathbb{F}^{\nu \times n}.$$

It consists of a token embedding, L transformer blocks, and a pointwise output layer. Thus $T(w)_{:,i} \in \mathbb{F}^{\nu}$ is defined at every ordinary position $1 \leq i \leq n$. We use $\mathcal{FPT}(L, D, H)$ for the class of such transformers with the corresponding parameter sizes.

Remark. All calculations below use the fixed-precision convention of Definition B.1.

Embedding. *A map $e : \Sigma \to \mathbb{F}^D$ determines the 0-th layer $\Lambda^{(0)}$, with*

$$\Lambda^{(0)}(w) \in \mathbb{F}^{D \times n}, \qquad \Lambda^{(0)}_{:,i}(w) = e(w_i) \quad (1 \leq i \leq n). \tag{12}$$

There are no positional encodings.

For $0 \leq \ell < L$ and $1 \leq h \leq H$, we define the attention mechanism of head h with input $\Lambda^{(\ell)} \in \mathbb{F}^{D \times n}$. We sometimes omit ℓ and h from the notation for readability purposes.

Attention. *An attention mechanism is a map $\mathrm{A}_h : \mathbb{F}^{D \times n} \to \mathbb{F}^{D_h \times n}$. First, choose $W^Q, W^K, W^V \in \mathbb{F}^{D_h \times D}$ and set*

$$Q = W^Q \Lambda^{(\ell)}, \qquad K = W^K \Lambda^{(\ell)}, \qquad V = W^V \Lambda^{(\ell)}. \tag{13}$$

The attention mechanism has a strict causal masking: the visible positions at i are $\{1, \ldots, i-1\}$. For $1 \leq j < i$, compute the attention score

$$S_{i,j} := \frac{Q^{\mathsf{T}}_{:,i} K_{:,j}}{\sqrt{D_h}}. \tag{14}$$

For a given position i, we consider the maximum of queried attention scores at position i $\max_i := \max_{j<i} S_{i,j}$ and define the stabilized attention weights

$$\alpha_{i,j} := \frac{\exp(S_{i,j} - \max_i)}{\sum_{k<i} \exp(S_{i,k} - \max_i)}.$$

The attention map is consequently defined as

$$\mathrm{A}_h(\Lambda^{(\ell)})_{:,i} := \sum_{j<i} \alpha_{i,j} V_{:,j}.$$

As there is no well-defined attention score for the first index $i = 1$, we set $\mathrm{A}_h(\Lambda^{(\ell)})_{:,1} := \mathbf{0} \in \mathbb{F}^{D_h}$.

We define by extension the head-agnostic attention mechanism $\mathrm{A}^{(\ell)}$ as the concatenation of the attention heads, with

$$\mathrm{A}^{(\ell)}_{:,i}(\Lambda^{(\ell)}) := \left(\mathrm{A}_1(\Lambda^{(\ell)})^{\mathsf{T}}_{:,i}, \ldots, \mathrm{A}_H(\Lambda^{(\ell)})^{\mathsf{T}}_{:,i}\right)^{\mathsf{T}}.$$

For each intermediate sublayer $\Lambda^{(\ell+0.5)}, 0 \leq \ell < L$, we define a position-wise feed-forward mechanism applied to this sublayer. We consider feed-forwards with one hidden layer of dimension D'. We set $D' = D$ for the base case, but it can be changed, handling the data complexity accordingly (c.f. Appendix **??** *We consider the* ReLU *activation function.*

Feed-forward. *Define* $\mathrm{FFN}_\ell : \mathbb{F}^D \to \mathbb{F}^D$*, applied independently at every one of the* n *positions, as*

$$\mathrm{FFN}_\ell(x) := \Theta_\ell^{F_2}\,\mathrm{ReLU}(\Theta_\ell^{F_1} x + b_\ell^{F_1}) + b_\ell^{F_2}, \qquad \mathrm{ReLU}(t) = \max(0, t),$$

where $\Theta_\ell^{F_1} \in \mathbb{F}^{D' \times D}$*,* $\Theta_\ell^{F_2} \in \mathbb{F}^{D \times D'}$*, and the biases have matching dimensions. The parameters are shared across all positions.*

Transformer block. *Transformer blocks are the combination of an attention mechanism with a feed-forward, together with residual streams and layer normalization* LN *(Ba et al., 2016) at each intermediate step. Each of the* L *transformer block is*

$$\Lambda^{(\ell+\frac{1}{2})} = \mathrm{LN}_\ell^{\mathrm{att}}\big(\mathrm{A}^{(\ell)}(\Lambda^{(\ell)}) + \Lambda^{(\ell)}\big), \tag{15}$$

$$\Lambda^{(\ell+1)} = \mathrm{LN}_\ell^{\mathrm{ff}}\big(\mathrm{FFN}_\ell(\Lambda^{(\ell+\frac{1}{2})}) + \Lambda^{(\ell+\frac{1}{2})}\big). \tag{16}$$

The layer normalization is applied column-wise, the two normalizations have separate parameters, and all parameters are shared across positions.

Output Layer *The output layer is the pointwise map* $\mathrm{O} : \mathbb{F}^D \to \mathbb{F}^\nu$ *given by*

$$\mathrm{O}(x) := \Theta^{\mathrm{out}} x + b^{\mathrm{out}}$$

where $\Theta^{\mathrm{out}} \in \mathbb{F}^{\nu \times D}$ *and* $b^{\mathrm{out}} \in \mathbb{F}^\nu$ *are the parameters shared across positions. Thus*

$$T(w)_{:,i} = \mathrm{O}\big(\Lambda_{:,i}^{(L)}(w)\big) \qquad (1 \le i \le n).$$

Attention arithmetic assumptions. For the transformers considered below, we impose the following conventions, motivated by the positive-sum argument of Li & Cotterell (2025, Appendix B.2, Lemma B.5). Attention sums depend only on the multiset of summands, and inserting zeros does not change them. Nonnegative summands have a nonnegative sum, with $+\infty$ absorbing. There is an integer $K_{\mathbb{F}} \ge 1$, independent of input length, such that any such sum containing more than $K_{\mathbb{F}}$ copies of a strictly positive finite value is $+\infty$. We write

$$r_{\mathbb{F}} \coloneqq K_{\mathbb{F}} + 1. \tag{17}$$

Stabilized exponentials are nonnegative; zero weights contribute zero even for infinite values. An empty prefix or a softmax denominator equal to 0 or $+\infty$ gives the zero head output. All remaining arithmetic cases are resolved deterministically.

C Logic

In this section, we define the past fragments of the linear temporal logic and of a two-variable first-order logic, introduced by Li & Cotterell (2025). These fragments are expressively equivalent and provide the link between temporal formulas and the partially-ordered structure studied below, that applies to transformers.

Fix a finite nonempty alphabet Σ and a word $w = w_1 \cdots w_n \in \Sigma^*$. The positions $1, \ldots, n$ are the ordinary token positions. We additionally use $n + 1$ as a *virtual logical endpoint*: it is an unlabelled position after the word, used only to evaluate word-level formulas.

C.1 Past temporal logic

First we introduce $\mathbf{LTL}[\mathbf{P}]$, the past fragment of the full linear temporal logic $\mathbf{LTL}[\mathbf{N}, \mathbf{U}, \mathbf{S}, \mathbf{P}]$ that provides several operators, such as *next*, *until*, and their past-time counterparts *since* and *past*. In our situation, the temporality refers to the positions of a given word, earlier positions thought as being in the past of later positions. The fragment $\mathbf{LTL}[\mathbf{P}]$ uses only the past operator $\mathbf{P}$, read "at some strictly earlier position". The restriction to the past operator reflects the strict masking of the attention mechanism. We use its pointed formulas to describe tokenwise transformer outputs, and evaluate word-level formulas at the virtual endpoint.

Definition C.1 (Syntax and semantics of $\mathbf{LTL}[\mathbf{P}]$). *The formulas of* $\boldsymbol{LTL}[\boldsymbol{P}]$ *are generated by*

$$\varphi ::= \top \mid \pi_a \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \mathbf{P}\varphi, \qquad a \in \Sigma. \tag{18}$$

For any position $1 \le i \le n+1$*, where* $n+1$ *is the virtual endpoint, satisfaction is defined inductively by*

$$\begin{aligned}
&w, i \models \top \quad \textit{always},\\
&w, i \models \pi_a \iff i \le n \textit{ and } w_i = a,\\
&w, i \models \neg\varphi \iff w, i \not\models \varphi,\\
&w, i \models \varphi \wedge \psi \iff w, i \models \varphi \textit{ and } w, i \models \psi,\\
&w, i \models \mathbf{P}\varphi \iff \exists j < i \textit{ with } w, j \models \varphi.
\end{aligned}$$

Thus $\mathbf{P}$ *means "at some strictly earlier position". The connectives* $\bot, \vee, \Rightarrow$ *are Boolean abbreviations. In particular,* $\neg\mathbf{P}\neg\varphi$ *means that* φ *holds at every earlier position, and is vacuously true at position* 1*.*

Definition C.2. *For a logic formula* $\Psi \in \boldsymbol{LTL}[\boldsymbol{P}]$*, set*

$$w \models \Psi := w, n+1 \models \Psi, \qquad L(\Psi) := \{w \in \Sigma^* : w \models \Psi\}. \tag{19}$$

Every atomic formula π_a*,* $a \in \Sigma$*, is false at the virtual endpoint. In particular, the formula* $\neg \bigvee_{a\in\Sigma} \pi_a$ *holds precisely at this endpoint. We also have* $L(\neg\mathbf{P}\top) = \{\epsilon\}$*.*

Example 1. For $a, b, c \in \Sigma$,

$$\begin{aligned}
L(\mathbf{P}\pi_a) &= \Sigma^* a \Sigma^*,\\
L\big(\mathbf{P}(\pi_c \wedge \mathbf{P}(\pi_b \wedge \mathbf{P}\pi_a))\big) &= \Sigma^* a \Sigma^* b \Sigma^* c \Sigma^*,\\
L\big(\mathbf{P}(\pi_a \wedge \neg\mathbf{P}\top)\big) &= a\Sigma^*.
\end{aligned}$$

The last formula selects an a with no predecessor, hence checking for first symbol. Note that we nevertheless can't express the language $\Sigma^* a$ in $\mathbf{LTL}[\mathbf{P}]$.

Lemma C.1 (Prefix invariance). *For* $u, v \in \Sigma^*$*,* $1 \le i \le |u|$*, and* $\varphi \in \boldsymbol{LTL}[\boldsymbol{P}]$*,*

$$u, i \models \varphi \quad \iff \quad uv, i \models \varphi.$$

Proof. Induct on φ. Atomic predicates depend only on the current symbol, and $\mathbf{P}$ quantifies only over positions strictly before i. □

C.2 Past two-variable first-order logic

The second logic that we introduce, a first-order logic, is equivalent to $\mathbf{LTL}[\mathbf{P}]$. The two-variable fragment $\mathbf{FO}^2[<]$ uses only the position variables x, y, which may be reused under quantifiers. We then further restrict to its past fragment $\mathbf{PFO}^2[<]$: each new occurrence of a free variable must be upper bounded by the other, again reflecting strict causal masking. We first give its pointed formulas, then its sentences. Naming two positions with the variables x and y is in particular useful for describing an attention query and a key.

The syntax mirrors the construction of $\mathbf{LTL}[\mathbf{P}]$: formulas are built from atomic statements using Boolean connectives and one restricted way of looking into the past.

Definition C.3 (Syntax and semantics of $\mathbf{PFO}^2[<]$). *For* $a \in \Sigma$ *and variables* x, y*, formulas are generated by*

$$\theta ::= \top \mid \pi_a(x) \mid \neg\theta \mid (\theta \wedge \theta) \mid \exists y < x : \theta(x, y).$$

As in Appendix C.1, positions range over $1, \ldots, n+1$*, with* $n+1$ *the virtual endpoint. For a current position* i *and, when needed, an earlier position* j*, satisfaction is defined inductively by*

$$\begin{aligned}
&w, i \models \top && \textit{always},\\
&w, i \models \pi_a(x) && \iff i \le n \textit{ and } w_i = a,\\
&w, i \models \neg\theta && \iff w, i \not\models \theta,\\
&w, i \models \theta \wedge \psi && \iff w, i \models \theta \textit{ and } w, i \models \psi,\\
&w, i \models \exists y < x : \theta(x, y) && \iff \exists j < i :\ w, i, j \models \theta(x, y).
\end{aligned}$$

A formula with one free variable is a pointed formula*; a formula with no free variables is a* sentence*. We use an unbounded existential quantifier only to close a pointed formula into a sentence, as in* $\exists x : \theta(x)$*. Boolean connectives and bounded universal quantifiers are abbreviations as usual.*

Example 2. The sentence

$$\exists x : \big(\pi_c(x) \wedge \exists y < x : \big(\pi_b(y) \wedge \exists x < y : \pi_a(x)\big)\big)$$

defines $\Sigma^* a \Sigma^* b \Sigma^* c \Sigma^*$. The innermost quantifier reuses x for the a-position: within its scope, the outer c-position is no longer named. This is the first-order analogue of $\mathbf{P}(\pi_c \wedge \mathbf{P}(\pi_b \wedge \mathbf{P}\pi_a))$ in $\mathbf{LTL}[\mathbf{P}]$.

Remark C.1. For every fixed $k \in \mathbb{N}$ and every one-variable formula $\theta(z)$, we can express that at least, or exactly, k earlier positions satisfy θ. Define recursively

$$\begin{aligned}
\exists^{\geq 0} y < x : \theta(y) &\coloneqq \top, \\
\exists^{\geq k+1} y < x : \theta(y) &\coloneqq \exists y < x : \big(\theta(y) \wedge \exists^{\geq k} x < y : \theta(x)\big), \\
\exists^{=k} y < x : \theta(y) &\coloneqq \big(\exists^{\geq k} y < x : \theta(y)\big) \wedge \neg\big(\exists^{\geq k+1} y < x : \theta(y)\big).
\end{aligned}$$

The recursive definition reuses x and y at each level, so it remains in $\mathbf{FO}^2[<]$. These are finite formulas for each fixed threshold k; they do not add an unbounded counting operation to the logic.

Lemma C.2 (Temporal–first-order correspondence)**.** *The logics $\mathbf{LTL}[\mathbf{P}]$ and $\mathbf{FO}^2[<]$ have the same expressive power (Li & Cotterell, 2025, Theorem 2.1). More precisely, every $\mathbf{LTL}[\mathbf{P}]$ formula is equivalent at each position to a $\mathbf{FO}^2[<]$ formula with one free variable, and every such $\mathbf{FO}^2[<]$ formula has an equivalent $\mathbf{LTL}[\mathbf{P}]$ formula; consequently, their sentences define the same class of languages. Under this correspondence, the basic one-position past test, where φ has only y free, is translated as*

$$\mathbf{P}\varphi \qquad \longleftrightarrow \qquad \exists y < x : \varphi(y).$$

C.3 Transformers as logic formulas

Under the attention arithmetic assumptions of Appendix B.2, fixed precision lets us describe every coordinate at every ordinary token position by a pointed $\mathbf{FO}^2[<]$ formula. This is the tokenwise version of the compilation of Li & Cotterell (2025); in particular, it does not use an `EOS` token.

Proposition C.1 (Transformer-to-logic)**.** *Let $T \in \mathcal{FPT}(L, D, H)$ have output dimension ν. For every output coordinate $1 \leq d \leq \nu$ and $f \in \mathbb{F}$, there is a pointed formula $\Phi^T_{d,f}(x) \in \mathbf{FO}^2[<]$ such that, for every $w \in \Sigma^*$ and every ordinary position $1 \leq i \leq |w|$,*

$$w, i \models \Phi^T_{d,f}(x) \quad \Longleftrightarrow \quad T(w)_{d,i} = f. \tag{20}$$

More generally, the same statement holds for every coordinate and value of each residual layer $\Lambda^{(\ell)}$.

Construction. For every layer ℓ, coordinate d, and $f \in \mathbb{F}$, construct $\Phi^{(\ell)}_{d,f}(x)$ so that

$$w, i \models \Phi^{(\ell)}_{d,f}(x) \quad \Longleftrightarrow \quad \Lambda^{(\ell)}_{d,i}(w) = f. \tag{21}$$

The embedding case is immediate:

$$\Phi^{(0)}_{d,f}(x) = \bigvee_{\substack{a \in \Sigma \\ e(a)_d = f}} \pi_a(x). \tag{22}$$

Every pointwise map $G : \mathbb{F}^m \to \mathbb{F}^r$ is a finite lookup table. If its input coordinates are described by $\varphi_{j,g}(x)$, its d-th output has value f precisely when

$$\Phi^G_{d,f}(x) \coloneqq \bigvee_{\substack{\mathbf{g} \in \mathbb{F}^m \\ G(\mathbf{g})_d = f}} \bigwedge_{j=1}^{m} \varphi_{j,g_j}(x). \tag{23}$$

Thus projections, residual connections, layer normalization, feed-forward maps, and the final output map preserve the construction using Boolean operations only.

It remains to consider one attention head. Its query, key, and value vectors are pointwise maps, so the preceding display gives formulas for every $\mathbf{q}, \mathbf{k}, \mathbf{v} \in \mathbb{F}^{D_h}$. Denote the corresponding formulas by

$$\mathrm{Q}^{(\ell,h)}_{\mathbf{q}}(x), \qquad \mathrm{K}^{(\ell,h)}_{\mathbf{k}}(y), \qquad \mathrm{V}^{(\ell,h)}_{\mathbf{v}}(y);$$

for example, $w, i \models \mathrm{Q}_{\mathbf{q}}^{(\ell,h)}(x)$ if and only if $Q_{:,i} = \mathbf{q}$. The relevant query–key/value relation is the two-variable formula

$$\theta_{\mathbf{q},\mathbf{k},\mathbf{v}}(x,y) \quad \equiv \quad \mathrm{Q}_{\mathbf{q}}^{(\ell,h)}(x) \wedge \mathrm{K}_{\mathbf{k}}^{(\ell,h)}(y) \wedge \mathrm{V}_{\mathbf{v}}^{(\ell,h)}(y), \qquad S(\mathbf{q},\mathbf{k}) = \frac{\mathbf{q}^{\mathsf{T}}\mathbf{k}}{\sqrt{D_h}}. \tag{24}$$

For each current query position x, the maximum score, softmax denominator, and weighted value sum are finite case distinctions over the multiplicities of earlier positions y satisfying these relations. In particular, the required fixed thresholds vary with the query position and are expressed directly in $\mathbf{PFO}^2[<]$ by

$$\exists^{\geq r} y < x : \theta_{\mathbf{q},\mathbf{k},\mathbf{v}}(x,y), \tag{25}$$

and their exact-count Boolean combinations, as in Remark C.1. Let $\Xi_{\mathbf{q},d,f}^{(\ell,h)}(x)$ denote the resulting finite Boolean combination that selects the cases in which head h outputs value f in coordinate d for query $\mathbf{q}$. Its output formula is

$$\Phi_{d,f}^{\mathrm{A},(\ell,h)}(x) = \bigvee_{\mathbf{q}\in\mathbb{F}^{D_h}} \left(\mathrm{Q}_{\mathbf{q}}^{(\ell,h)}(x) \wedge \Xi_{\mathbf{q},d,f}^{(\ell,h)}(x) \right). \tag{26}$$

Fixed precision gives finitely many query cases, and the arithmetic assumptions of Appendix B.2 let us clip multiplicities at $r_{\mathbb{F}}$. Thus only finitely many cases are needed.

Consequently, attention produces formulas satisfying equation 21 for the next sublayer, and the pointwise steps complete the induction over ℓ. Applying the final output lookup formula equation 23 yields equation 20. □

All formulas in Proposition C.1 are evaluated at ordinary token positions. By Lemma C.2, each has an equivalent pointed $\mathbf{LTL}[\mathbf{P}]$ formula, but the PFO presentation keeps the current position x and the earlier position y explicit.

C.4 Partially ordered automata

Word-level formulas define languages, whereas the pointed formulas equation 20 describe an output at each token. The corresponding automata verifying the formulas are respectively DFAs and Mealy machines.

Definition C.4 (Deterministic Finite Automaton)**.** *A* DFA *is a tuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, F_A)$, with finite state set Q, initial state q_0, accepting set $F_A \subseteq Q$, and transition function $\delta : Q \times \Sigma \to Q$. By abuse of notation, extend δ to words by iterating over their letters:*

$$\delta(q, \epsilon) = q, \qquad \delta(q, ua) = \delta(\delta(q, u), a).$$

It recognises the language $L(\mathcal{A}) = \{u : \delta(q_0, u) \in F_A\}$.

For our results, we do not need the following Mealy machine construction, as we work with for the difference function that gets expressed as a poDFA. Nevertheless, Mealy machines closely matches our transformer definition: both process the input left to right and emit an output vector at every position from the past prefix and the current token. We shall see during next proof developments to bound the height of the difference function, that some arguments can directly be applied on individual Mealy machines. This provides the opportunity to deepen the ordered-automaton analyses of the transformers outside of our student–teacher learning framework.

Definition C.5 (Mealy Machine)**.** *A* Mealy machine *$\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, \lambda)$ is an automaton in the same way as the DFA C.4, but has a finite output alphabet Γ, an output map $\lambda : Q \times \Sigma \to \Gamma$ and not notion of accepting subset of states. On input $w = w_1 \cdots w_n$, it emits*

$$\mathcal{M}(w)_i = \lambda(\delta(q_0, w_{1:i-1}), w_i) \qquad (1 \leq i \leq n).$$

Intuitively, Mealy machines are automata such that their restricted behaviour, with respect to one output symbol, resembles a DFA.

Definition C.6 (Partially ordered automata)**.** *A DFA $\mathcal{A}$ or a Mealy machine $\mathcal{M}$, is said to be* partially ordered *if reachability of its states is antisymmetric, equivalently if it contains no cycles but self-loops. Its* height *is the largest number of strict state changes along a run:*

$$\mathrm{ht}(\mathcal{M}) = \max_{w\in\Sigma^*} \big|\{i : \delta(q_0, w_{1:i}) \neq \delta(q_0, w_{1:i-1})\}\big|,$$

with the same definition for $\mathcal{A}$. It is reduced *if all states are reachable and if two distinct states have different future behaviour.*

Remark. Different future behaviours for DFAs means that for any two distinct states representers w, w', there exists a suffix $v \in \Sigma^*$ such that, w.l.o.g. $\delta(w, v) \in F_A$ while $\delta(w', v) \notin F_A$. For Mealy machines, it similarly means that for some suffix concatenation, the future emitted sequences must disagree.

We recall the language-level correspondence established by Li & Cotterell (2025, Section 4 and Appendix C).

Proposition C.2. *For every $\Psi \in \boldsymbol{LTL}[\boldsymbol{P}]$, the language $L(\Psi) = \{w : w, |w| + 1 \models \Psi\}$ is regular and its reduced DFA is partially ordered. The same holds for the equivalent word-level $\boldsymbol{PFO}^2[<]$ formulas.*

Proof. Li and Cotterell show that the syntactic monoid of an $\mathbf{LTL}[\mathbf{P}]$-definable language is $\mathcal{R}$-trivial (Li & Cotterell, 2025, Appendix C): distinct monoid elements have distinct principal right ideals. The characterisation of Brzozowski & Fich (1980) identifies these regular languages exactly with those whose reduced DFA is partially ordered. The $\mathbf{PFO}^2[<]$ statement follows from Lemma C.2. □

The states of this automaton have the usual right-context interpretation given by Myhill–Nerode (Myhill, 1957; Nerode, 1958).

Proposition C.3. *Let $\mathcal{A}$ be the reduced DFA of a regular language $\mathcal{L}$. For any $u, v \in \Sigma^*$,*

$$\delta(q_0, u) = \delta(q_0, v) \iff \forall z \in \Sigma^* : \quad (uz \in \mathcal{L} \iff vz \in \mathcal{L}).$$

Thus its states are precisely the classes of prefixes with the same accepting continuations. For $\mathcal{L} = L(\Psi)$, these classes form the states of the reduced poDFA above.

Proof. Prefixes reaching the same state have the same acceptance decision on every continuation, by determinism. Conversely, two distinct states of a reduced DFA admit a distinguishing continuation. Every state is reachable, so this identifies the states with the right-context classes. □

We now apply this correspondence to the tokenwise transformer formulas 20. Their past tests are recognised by poDFAs, and the current letter determines the output together with the states of these automata.

Proposition C.4 (Transformers as partially-ordered Mealy machines)**.** *Every transformer admitting the formulas equation 20 is realised by a finite partially ordered Mealy machine with output alphabet $\Gamma = \mathbb{F}^\nu$:*

$$\mathcal{M}_T(w)_i = T(w)_{:,i} \qquad (1 \le i \le |w|).$$

In particular, Proposition C.1 provides these formulas for the transformers of Appendix B.2.

Proof. Translate the pointed $\mathbf{PFO}^2[<]$ formulas into equivalent $\mathbf{LTL}[\mathbf{P}]$ formulas $\Psi^T_{d,f}$ by Lemma C.2, and let $\mathcal{P}$ collect their distinct past subformulas $\chi = \mathbf{P}\psi$. For each χ, Proposition C.2 provides a reduced poDFA $\mathcal{A}_\chi = (Q_\chi, \Sigma, \delta_\chi, q_{0,\chi}, F_\chi)$ recognising $L(\chi)$. After a prefix u, its acceptance bit gives exactly the corresponding past test at the next token:

$$\delta_\chi(q_{0,\chi}, u) \in F_\chi \iff u\sigma, |u| + 1 \models \chi \qquad (\sigma \in \Sigma).$$

Indeed, both sides ask whether ψ holds at some position in u; prefix invariance preserves those truth values.

Take the reachable part of the product of these poDFAs, with

$$q_0 = (q_{0,\chi})_{\chi\in\mathcal{P}}, \qquad \delta(q, \sigma) = \big(\delta_\chi(q_\chi, \sigma)\big)_{\chi\in\mathcal{P}}.$$

For each current letter σ, evaluate $\Psi_{d,f}^T$ by replacing π_a with $\mathbf{1}\{\sigma = a\}$ and each past test χ with $\mathbf{1}\{q_\chi \in F_\chi\}$, respecting Boolean connectives. Set $\lambda(q, \sigma)_d$ to the unique f for which this evaluation is true. Equation equation 20 ensures existence and uniqueness on reachable states. The output uses the state before reading σ, so the past remains strict. Finally, every component can only advance in its reachability order; their finite product is therefore partially ordered. If $\mathcal{P}$ is empty, the product has one state. □

This proposition establishes the inherent ordered structure of the transformers, using Mealy machines. We shall see quickly that the difference function between two transformers translates naturally in the other ordered automaton that we defined earlier: the poDFA.

Remark C.2. Each $\Phi_{d,f}^T(x)$ gives a Boolean output on transitions of this machine. Testing it at the last ordinary token need not define a poDFA language: $\pi_a(x)$ would give $\Sigma^* a$, whose reduced DFA has a nontrivial cycle when $|\Sigma| \geq 2$. The poDFAs used above instead recognise past tests at the virtual endpoint. Importantly, a fixed Mealy state may still emit different vectors on different input letters.

To identify the states intrinsically, write

$$T_u(z) \coloneqq \big(T(uz)_{:,|u|+j}\big)_{j=1}^{|z|}.$$

This is the sequence emitted on the continuation z after reading u; it excludes the outputs already emitted on u.

Proposition C.5. *For a transformer satisfying Proposition C.4, there is a unique reduced Mealy machine, up to isomorphism. Its states are the classes of*

$$u \sim_T v \quad \Longleftrightarrow \quad \forall z \in \Sigma^* : \; T_u(z) = T_v(z),$$

and its transitions and outputs are

$$q_0 = [\epsilon], \qquad \delta([u], a) = [ua], \qquad \lambda([u], a) = T(ua)_{:,|u|+1}.$$

This reduced machine is partially ordered. Consequently,

$$\delta(q_0, u) = \delta(q_0, v) \quad \Longleftrightarrow \quad \forall z \in \Sigma^* : \; T_u(z) = T_v(z).$$

Proof. Causality gives $T_u(az) = T_u(a)\, T_{ua}(z)$, with concatenation of output sequences. Hence $u \sim_T v$ implies equal outputs on a and $ua \sim_T va$, so the displayed maps are well defined. Equal states of the finite machine in Proposition C.4 imply equivalent prefixes; our construction is therefore a finite quotient. All classes are reachable and distinct classes have a distinguishing continuation. In any reduced realisation, these same classes correspond bijectively to states, proving reducedness and uniqueness.

Finally, reduction preserves partial order: repeatedly traversing a nontrivial quotient cycle would force infinitely many state changes in the original finite partially ordered machine, a contradiction. □

C.5 Past subformulas and automaton height

We relate past subformulas to the height of their associated automata. Word satisfaction is always evaluated at the virtual endpoint $|w| + 1$ of Appendix C.1.

Definition C.7 (Subformulas)**.** *For $\Psi \in$ **LTL**[**P**], let $\mathrm{Sub}(\Psi)$ be the set of its syntactically distinct subformulas, including Ψ itself. Recursively,*

$$\begin{aligned} \mathrm{Sub}(\top) &= \{\top\}, & \mathrm{Sub}(\pi_a) &= \{\pi_a\}, \\ \mathrm{Sub}(\neg\varphi) &= \{\neg\varphi\} \cup \mathrm{Sub}(\varphi), & \mathrm{Sub}(\mathbf{P}\varphi) &= \{\mathbf{P}\varphi\} \cup \mathrm{Sub}(\varphi), \\ \mathrm{Sub}(\varphi \wedge \psi) &= \{\varphi \wedge \psi\} \cup \mathrm{Sub}(\varphi) \cup \mathrm{Sub}(\psi). \end{aligned}$$

The past subformulas *are those whose outermost operator is* $\mathbf{P}$*:* $\mathrm{Sub}_{\mathbf{P}}(\Psi) \coloneqq \{\mathbf{P}\varphi : \mathbf{P}\varphi \in \mathrm{Sub}(\Psi)\}$.

Definition C.8 (Past profile)**.** *For any formula $\Psi \in \textbf{\textit{LTL}}[\boldsymbol{P}]$, we define the past profile of a word $u \in \Sigma^*$ as the vector*

$$\mathrm{prof}_\Psi(u) \in \{0,1\}^{\mathrm{Sub}_{\mathbf{P}}(\Psi)}, \qquad \mathrm{prof}_\Psi(u)(\mathbf{P}\varphi) = \mathbf{1}\{u, |u|+1 \models \mathbf{P}\varphi\}. \tag{27}$$

It records which past subformulas hold after reading u, at its virtual endpoint $|u|+1$.

Example 3. For $\Psi = \mathbf{P}(\pi_b \wedge \mathbf{P}\pi_a)$, order the profile coordinates as $(\mathbf{P}\pi_a, \Psi)$. For distinct a, b, the profiles of ϵ, a, ab are respectively $(0,0), (1,0), (1,1)$.

For a language $L \subseteq \Sigma^*$, we note the left residual at a prefix u with respect to a language L as

$$u^{-1}L \coloneqq \{v \in \Sigma^* : uv \in L\}.$$

Lemma C.3. *Let $u_1, u_2 \in \Sigma^*$. If*

$$\mathrm{prof}_\Psi(u_1) = \mathrm{prof}_\Psi(u_2), \tag{PR}$$

then,

$$u_1^{-1}L(\Psi) = u_2^{-1}L(\Psi);$$

equivalently for every $v \in \Sigma^*$,

$$u_1 v \in L(\Psi) \quad \Longleftrightarrow \quad u_2 v \in L(\Psi).$$

Proof. Write $l_i = |u_i|$ and $m = |v|$. By structural induction on $\varphi \in \mathrm{Sub}(\Psi)$, we show that, for $0 \le k \le m$,

$$u_1 v, l_1 + k + 1 \models \varphi \quad \Longleftrightarrow \quad u_2 v, l_2 + k + 1 \models \varphi. \tag{28}$$

First, the atomic case compares the same suffix symbol, or the two virtual endpoints; hence the equivalence holds. The Boolean cases follow immediately.

Now consider $\varphi = \mathbf{P}\theta$. For $i \in \{1,2\}$,

$$u_i v, l_i + k + 1 \models \mathbf{P}\theta \iff \mathrm{prof}_\Psi(u_i)(\mathbf{P}\theta) = 1 \ \vee \ \exists\, 0 \le j < k : \ u_i v, l_i + j + 1 \models \theta.$$

The first case says that θ already has a witness in u_i, which is stored in the profile. The second says that it has a witness in the first k positions of v. By (PR), the first cases agree; by the induction hypothesis, so do the second. Thus the equivalence holds. Taking $\varphi = \Psi$ and $k = m$ proves the claim. □

Corollary C.1. *Let $\Psi \in \textbf{\textit{LTL}}[\boldsymbol{P}]$ and let A_Ψ be its minimal DFA. If two prefixes $u_1, u_2 \in \Sigma^*$ have the same past profile, then they reach the same state of A_Ψ.*

Proof. Lemma C.3 gives us that for every suffix $v \in \Sigma^*$,

$$u_1 v \in L(\Psi) \quad \Longleftrightarrow \quad u_2 v \in L(\Psi).$$

This is precisely the characterisation of Proposition C.3. □

Definition C.9. *For $b, b' \in \{0,1\}^{\mathrm{Sub}_{\mathbf{P}}(\Psi)}$, write $b \preceq b'$ when*

$$\forall \chi \in \mathrm{Sub}_{\mathbf{P}}(\Psi), \qquad b(\chi) \le b'(\chi).$$

Write $b \prec b'$ when $b \preceq b'$ and $b \neq b'$.

Lemma C.4. *For all $u, v \in \Sigma^*$,*

$$\mathrm{prof}_\Psi(u) \preceq \mathrm{prof}_\Psi(uv).$$

Proof. Let $\mathbf{P}\theta \in \mathrm{Sub}_{\mathbf{P}}(\Psi)$ and suppose that $\mathrm{prof}_\Psi(u)(\mathbf{P}\theta) = 1$. By definition, there is a position $m < |u| + 1$ such that $u, m \models \theta$. By Lemma C.1, $uv, m \models \theta$; also $m < |uv| + 1$. Hence $uv, |uv| + 1 \models \mathbf{P}\theta$, so $\mathrm{prof}_\Psi(uv)(\mathbf{P}\theta) = 1$. This holds for every coordinate. □

Lemma C.5. *Let S be a finite set and let*

$$b_0 \prec b_1 \prec \cdots \prec b_r \qquad \textit{in } \{0,1\}^S$$

under the coordinate-wise order. Then $r \le |S|$.

Proof. For every i, let $S_i \subseteq S$ be the support of b_i, namely the set of coordinates on which b_i is 1. The strict inequalities give

$$S_0 \subsetneq S_1 \subsetneq \cdots \subsetneq S_r \subseteq S.$$

Every strict inclusion increases cardinality by at least one. Thus there are at most $|S|$ strict inclusions, proving $r \leq |S|$. □

Lemma C.6. *Let $u_0, \ldots, u_r \in \Sigma^*$ be a sequence of prefixes such that, for every $0 \leq i < r$, there is a word $x_i \in \Sigma^*$ satisfying*

$$u_{i+1} = u_i x_i \qquad \textit{and} \qquad u_i^{-1} L(\Psi) \neq u_{i+1}^{-1} L(\Psi).$$

Then

$$r \leq \left|\mathrm{Sub}_{\mathbf{P}}(\Psi)\right|.$$

Proof. By Lemma C.4,

$$\mathrm{prof}_\Psi(u_i) \preceq \mathrm{prof}_\Psi(u_{i+1}) \qquad (0 \leq i < r).$$

If two consecutive profiles were equal, Lemma C.3 would imply $u_i^{-1} L(\Psi) = u_{i+1}^{-1} L(\Psi)$, contrary to assumption. Hence each displayed relation is strict. Lemma C.5, applied with $S = \mathrm{Sub}_{\mathbf{P}}(\Psi)$, gives the result. □

We finally combine the previous definitions and results for the following result. In particular, the bound C_0 from Eq. 2 is an application of this theorem:

Theorem C.1 (Past subformulas bound automaton height)**.** *For $\Psi \in \textbf{\textit{LTL}}[\boldsymbol{P}]$, the reduced DFA $\mathcal{A}_\Psi$ of $L(\Psi)$ satisfies*

$$\mathrm{ht}(\mathcal{A}_\Psi) \leq |\,\mathrm{Sub}_{\mathbf{P}}(\Psi)|.$$

Proof. The bound follows from Lemma C.6. □

We can directly convey the above result to the Mealy machine. Intuitively, our transformers from Appendix B.2 are the Mealy machines made of the pointed formulas given in 20.

Corollary C.2. *For a finite family $\mathcal{G}$ of pointed $\textbf{\textit{LTL}}[\boldsymbol{P}]$ formulas, the reduced Mealy machine $\mathcal{M}_\mathcal{G}$ emitting their truth values at each token similarly satisfies*

$$\mathrm{ht}(\mathcal{M}_\mathcal{G}) \leq \left|\bigcup_{\varphi \in \mathcal{G}} \mathrm{Sub}_{\mathbf{P}}(\varphi)\right|.$$

Proof. For the Mealy bound, store the shared past profile over $\bigcup_{\varphi\in\mathcal{G}} \mathrm{Sub}_{\mathbf{P}}(\varphi)$. Together with the next letter, it determines the current truth values and the updated profile: the bit for $\mathbf{P}\theta$ becomes its old value OR the current truth of θ. Thus these profiles form a Mealy realisation. Each bit changes at most once, giving the stated height; reduction cannot increase it. □

D Length Generalization

For two FPTs T_s, T_t sharing Σ, $\mathbb{F}$, and output dimension ν, the *difference function* from Section 3.1 is

$$\Delta(w) \coloneqq \mathbf{1}\{\exists\, 1 \leq i \leq |w| :\ T_s(w)_{:,i} \neq T_t(w)_{:,i}\}. \tag{29}$$

We give two height bounds for its reduced poDFA $\mathcal{A}_\Delta$. We use the arithmetic assumptions of Appendix B.2, with $p = |\mathbb{F}|$ and $r_\mathbb{F} = K_\mathbb{F} + 1$.

D.1 BOUNDING THE HEIGHT WITH PAST SUBFORMULAS

We count the past subformulas needed to describe one transformer's outputs, then combine the two families into a formula for the difference function.

Lemma D.1 (Simultaneous output representation)**.** *Under these arithmetic hypotheses, let $T \in \mathcal{FPT}(L, D, H)$ have output dimension ν, and put $D_h = D/H$. There are formulas $\Phi^T_{d,f} \in \boldsymbol{LTL}[\boldsymbol{P}]$, for $1 \le d \le \nu$ and $f \in \mathbb{F}$, such that, for every $w \in \Sigma^*$ and $1 \le i \le |w|$,*

$$w, i \models \Phi^T_{d,f} \quad \Longleftrightarrow \quad T(w)_{d,i} = f.$$

Their shared collection of distinct past subformulas satisfies

$$\mathcal{P}(T) \coloneqq \bigcup_{d=1}^{\nu} \bigcup_{f \in \mathbb{F}} \mathrm{Sub}_{\mathbf{P}}(\Phi^T_{d,f}), \qquad |\mathcal{P}(T)| \le B(T) \coloneqq r_{\mathbb{F}} L D \, p^{D/H+1}. \tag{30}$$

Proof. Follow the layerwise construction of Appendix C.3, now directly in $\mathbf{LTL}[\mathbf{P}]$. At layer 0, the embedding coordinates are described by

$$\Phi^{(0)}_{d,f} = \bigvee_{a \,:\, e(a)_d = f} \pi_a.$$

Moreover, if a pointwise map g sends input coordinates with values $(f_1, \dots, f_r)$ to $g(f_1, \dots, f_r)$, then its output value f is described by the Boolean combination

$$\bigvee_{(f_1,\dots,f_r) \,:\, g(f_1,\dots,f_r) = f} \; \bigwedge_{j=1}^{r} \Phi_{j,f_j}.$$

Thus projections, feed-forward maps, residuals, normalization, and the output map add no past operators. Only attention does.

Fix a layer ℓ and a head h, suppressing these indices below. For each key vector $\mathbf{k} \in \mathbb{F}^{D_h}$, coordinate $d \le D_h$, and value $f \in \mathbb{F}$, let $\theta_{\mathbf{k},d,f}$ describe the unary condition $K_{:,i} = \mathbf{k}$ and $V_{d,i} = f$. These pointwise predicates add no past operators. Writing $\theta(y)$ for their unary $\mathbf{PFO}^2[<]$ form, we describe the clipped counts

$$c_{\mathbf{k},d,f}(i) = \min\bigl\{r_{\mathbb{F}}, \bigl|\{j < i : K_{:,j} = \mathbf{k},\ V_{d,j} = f\}\bigr|\bigr\}$$

using the threshold tests

$$\exists^{=r} y < x : \theta(y) \quad (0 \le r < r_{\mathbb{F}}), \qquad \exists^{\ge r_{\mathbb{F}}} y < x : \theta(y). \tag{31}$$

By Remark C.1, each successive at-least threshold adds one outer $\mathbf{P}$ to the conjunction of θ with the previous threshold. Thresholds $1, \dots, r_{\mathbb{F}}$ share their shorter expansions, and exact counts are Boolean combinations of successive thresholds (threshold 0 is $\top$). Hence all tests for one θ add at most $r_{\mathbb{F}}$ distinct past subformulas.

Fix an output coordinate d. The current query $\mathbf{q} = Q_{:,i}$ and the clipped-count vector

$$\mathbf{c}_i = \bigl(c_{\mathbf{k},d,f}(i)\bigr)_{\mathbf{k} \in \mathbb{F}^{D_h},\ d \le D_h,\ f \in \mathbb{F}}$$

determine the head output. Indeed, these data identify the present keys, the maximum score, and each stabilized exponential $a_{\mathbf{k}}$. If some $a_{\mathbf{k}} = +\infty$, or if

$$c_{\mathbf{k},d,f}(i) = r_{\mathbb{F}} \quad \text{and} \quad a_{\mathbf{k}} > 0,$$

then the denominator overflows and the head output is zero. Otherwise, every type with a positive exponential has its exact multiplicity recorded, while zero terms contribute nothing. Multiset summation then determines the denominator, the weights, and the weighted value sum. Hence, for a finite lookup function A_d,

$$\mathrm{Att}_{d,i} = A_d(\mathbf{q}, \mathbf{c}_i).$$

Expanding this finite lookup into Boolean cases reuses only the threshold tests above, and so introduces no further past subformulas.

There are

$$|\mathbb{F}^{D_h}|\, D_h\, |\mathbb{F}| = D_h p^{D_h+1}$$

key–coordinate–value types per head. Thus, if $\mathcal{P}_\ell$ denotes the past subformulas available after layer ℓ, then

$$|\mathcal{P}_\ell \setminus \mathcal{P}_{\ell-1}| \le r_{\mathbb{F}} H D_h p^{D_h+1} = r_{\mathbb{F}} D p^{D/H+1}.$$

Summing this bound over the L layers proves equation 30. All ν output coordinates and all $f \in \mathbb{F}$ reuse the same collection, so the final output map adds no factor. □

Remark D.1. We count distinct past subformulas, not occurrences or Boolean formula size. A general $\exists y < x : \theta(x, y)$ need not translate into a single $\mathbf{P}$-subformula. Here all counted predicates depend only on y; the current query is handled outside the past operators by Boolean cases.

The predicates $\Phi^T_{d,f}$ describe the outputs of the Mealy machines from Appendix C.4. Comparing these predicates and checking for a mismatch anywhere in the word gives the difference function.

Proposition D.1 (Difference function as a logic formula)**.** *Let T_s, T_t be student and teacher transformers satisfying the preceding hypotheses, over the same alphabet and precision set, with the same output dimension ν. Using the formulas of Lemma D.1, define*

$$\psi_\Delta \coloneqq \bigvee_{d=1}^{\nu} \bigvee_{f\in\mathbb{F}} \big(\Phi^{T_s}_{d,f} \wedge \neg \Phi^{T_t}_{d,f}\big), \tag{32}$$

$$\Psi_\Delta \coloneqq \mathbf{P}\psi_\Delta. \tag{33}$$

Then

$$w \models \Psi_\Delta \quad \Longleftrightarrow \quad \Delta(w) = 1,$$

and

$$|\operatorname{Sub}_{\mathbf{P}}(\Psi_\Delta)| \le B(T_s) + B(T_t) + 1. \tag{34}$$

Moreover, $L(\Psi_\Delta)\Sigma^* = L(\Psi_\Delta)$.

Proof. At each token, ψ_Δ holds exactly when an output coordinate differs. Thus $\mathbf{P}\psi_\Delta$, evaluated at $|w|+1$, expresses $\Delta(w) = 1$ (and is false on ϵ). Boolean operations add no past subformulas:

$$\operatorname{Sub}_{\mathbf{P}}(\Psi_\Delta) \subseteq \mathcal{P}(T_s) \cup \mathcal{P}(T_t) \cup \{\mathbf{P}\psi_\Delta\}.$$

This proves the count. Causality preserves an existing mismatch under right concatenation, proving persistence. □

Corollary D.1. *The reduced DFA $\mathcal{A}_\Delta$ of $L(\Psi_\Delta)$ is partially ordered and satisfies*

$$\operatorname{ht}(\mathcal{A}_\Delta) \le B(T_s) + B(T_t) + 1 = r_{\mathbb{F}}\big(L_s D_s p^{D_s/H_s+1} + L_t D_t p^{D_t/H_t+1}\big) + 1.$$

If Δ is not identically zero, $\mathcal{A}_\Delta$ has a unique accepting state, which is absorbing.

Proof. Apply Proposition C.2 and Theorem C.1 to Ψ_Δ, then use equation 34. Every accepted prefix has residual Σ^* by persistence, so all accepting prefixes reach the same absorbing state by Proposition C.3. □

Remark D.2 (Mealy-machine interpretation)**.** Let $\mathcal{M}_T$ be the reduced Mealy machine of T. The shared-profile argument of Appendix C.5 gives $\operatorname{ht}(\mathcal{M}_T) \le B(T)$. Running the two machines in parallel until their outputs differ gives

$$\operatorname{ht}(\mathcal{A}_\Delta) \le \operatorname{ht}(\mathcal{M}_{T_s}) + \operatorname{ht}(\mathcal{M}_{T_t}) + 1. \tag{35}$$

Each strict product transition advances at least one component; entering the absorbing bad state adds at most one. Reduction cannot increase height.

D.2 BOUNDING THE HEIGHT WITH TEMPORAL PROFILES

We bound the same difference-function poDFA $\mathcal{A}_\Delta$ using past nesting depth. This removes the exponential dependence on D/H and gives a polynomial bound in depth, of degree $s = |\Sigma|$.

Definition D.1 (Past depth and temporal profiles)**.** *The* past depth *of a formula is defined by*

$$\begin{aligned} \mathrm{pd}(\top) = \mathrm{pd}(\pi_a) = 0, \qquad & \mathrm{pd}(\neg\varphi) = \mathrm{pd}(\varphi), \\ \mathrm{pd}(\mathbf{P}\varphi) = 1 + \mathrm{pd}(\varphi), \qquad & \mathrm{pd}(\varphi \wedge \psi) = \max\{\mathrm{pd}(\varphi), \mathrm{pd}(\psi)\}. \end{aligned}$$

For $w \in \Sigma^*$, write $w_{<i} = w_1 \cdots w_{i-1}$ and define

$$U_0(w) = \star, \qquad U_{k+1}(w) = \big\{(w_i, U_k(w_{<i})) : 1 \le i \le |w|\big\}, \tag{36}$$

where $\star$ is a fixed symbol. Unlike prof_Ψ, the profile U_k does not depend on a particular formula: it records which (letter, lower-depth-profile) pairs have occurred.

The profile records which letters have appeared in which past contexts. For fixed k, there are finitely many profiles, and higher profiles determine all lower ones by forgetting their deepest context layer. For $k \ge 1$,

$$U_k(ua) = U_k(u) \cup \{(a, U_{k-1}(u))\}. \tag{37}$$

Thus profiles grow by inclusion and update deterministically from the next letter. Structural induction shows that $(w_i, U_k(w_{<i}))$ determines every formula of past depth at most k at position i. At the virtual endpoint, all letter atoms are false, so $U_k(w)$ alone determines truth. Equal profiles therefore give equal residual languages.

Lemma D.2 (Counting strict profile changes)**.** *For every $k \ge 0$, along the prefixes of any word, U_k changes at most*

$$\binom{k+s}{s} - 1 \tag{38}$$

times. Consequently, if $\mathrm{pd}(\Psi) \le k$, the reduced DFA of $L(\Psi)$ has height at most $\binom{k+s}{s} - 1$.

Proof. For each $a \in \Sigma$, define its next-change rank after u by

$$r_a(u) = \min\{j \ge 1 : U_j(ua) \ne U_j(u)\}.$$

This is finite because $\mathbf{P}^{|u|+1}\top$ distinguishes u from ua at the virtual endpoint. In particular,

$$U_k(ua) \ne U_k(u) \quad \Longleftrightarrow \quad r_a(u) \le k. \tag{39}$$

Initially, $r_a(\epsilon) = 1$. If $t = r_a(u)$, the ranks update as

$$r_a(ua) = t + 1, \qquad r_b(ua) = \min\{r_b(u), t+1\} \quad (b \ne a). \tag{40}$$

A letter not yet seen has rank 1. Otherwise, comparing with the prefix before its last occurrence shows that its rank is one plus the first depth at which the intervening profiles differ. Appending a changes profiles from depth t onwards, which gives equation 40.

For fixed k, let

$$\rho_k(u) \coloneqq \mathrm{sort}\big(\min\{r_a(u), k+1\} : a \in \Sigma\big).$$

If $t \le k$, Equation equation 40 replaces one entry t by $t+1$, leaves all entries below t fixed, and leaves every other altered entry at least $t+1$. Hence

$$U_k(ua) \ne U_k(u) \quad \Longrightarrow \quad \rho_k(u) <_{\mathrm{lex}} \rho_k(ua).$$

If $t > k$, both U_k and ρ_k are unchanged. Since there are exactly

$$\binom{k+s}{s}$$

nondecreasing s-tuples in $\{1, \ldots, k+1\}^s$, the profile changes at most $\binom{k+s}{s} - 1$ times. The profile automaton recognises $L(\Psi)$; reduction cannot increase its height. □

This counts profile changes along a run, not the total number of profiles.

Lemma D.3 (Past depth of transformer outputs)**.** *Under the stated assumptions, a transformer $T \in \mathcal{FPT}(L, D, H)$ with output dimension ν admits coordinate/value formulas $\Phi^T_{d,f} \in \boldsymbol{LTL}[\boldsymbol{P}]$ satisfying*

$$w, i \models \Phi^T_{d,f} \iff T(w)_{d,i} = f, \qquad \mathrm{pd}(\Phi^T_{d,f}) \le r_{\mathbb{F}} L,$$

for every $1 \le i \le |w|$, $1 \le d \le \nu$, and $f \in \mathbb{F}$.

Proof. In the construction of Lemma D.1, embeddings have depth zero and pointwise maps do not increase depth. At each layer, the unary key/value predicates inherit the previous layer's depth. Counting their occurrences up to $r_{\mathbb{F}}$ adds at most $r_{\mathbb{F}}$ nested past operators; attention then uses only Boolean combinations of these tests and the current query predicates. Each layer therefore increases depth by at most $r_{\mathbb{F}}$. □

Corollary D.2. *Put $L_* = \max\{L_s, L_t\}$. The reduced poDFA $\mathcal{A}_\Delta$ of the difference function satisfies*

$$\mathrm{ht}(\mathcal{A}_\Delta) \le \binom{r_{\mathbb{F}} L_* + s}{s}. \tag{41}$$

Proof. Set $k = r_{\mathbb{F}} L_*$. Using the output formulas of Lemma D.3 in equation 32 gives $\mathrm{pd}(\psi_\Delta) \le k$. Construct a DFA that stores $U_k(u)$ until the first mismatch. On reading a, $(a, U_k(u))$ determines ψ_Δ at the new position: if true, enter an accepting absorbing state; otherwise update U_k. This recognises $L(\Psi_\Delta)$.

The profile grows by inclusion, so the DFA is partially ordered. Lemma D.2 allows at most $\binom{k+s}{s} - 1$ profile changes, and entering the accepting state adds at most one. Reduction preserves partial order and cannot increase height. □

For fixed Σ and $\mathbb{F}$, this bound is polynomial in L_*, of degree $s = |\Sigma|$, and independent of widths, head counts, and output dimension. It bounds automaton height, not the size of the representing formulas.

Remark D.3 (Mealy-machine interpretation)**.** The same profile $U_{r_{\mathbb{F}} L}(u)$, together with the next letter, determines all outputs of T. It therefore also realises T as a Mealy machine, giving

$$\mathrm{ht}(\mathcal{M}_T) \le \binom{r_{\mathbb{F}} L + s}{s} - 1. \tag{42}$$

Summing the two individual bounds via equation 35 also bounds $\mathrm{ht}(\mathcal{A}_\Delta)$. The poDFA proof above is sharper because both transformers share $U_{r_{\mathbb{F}} L_*}$ on their common input.

Remark D.4. We remark that Masopust & Thomazo (2015) showed a related bound (also in terms of a binomial coefficient) on the height of the minimal DFA representing piecewise testable languages; such languages are expressible by fixed-precision transformers. Our result differs in applying to difference automata for general fixed-precision transformers, and in expressing the bound in terms of the size parameters of those transformers.

D.3 Deterministic length generalization

Either height bound from Corollary D.1-D.2 gives a deterministic length cutoff: any nonzero difference function between two transformers has its behaviour controlled by height of its poDFA. Taking the smaller bound yields the following result.

Lemma D.4 (Deterministic length generalization)**.** *Let $T_s \in \mathcal{FPT}(L_s, D_s, H_s)$ and $T_t \in \mathcal{FPT}(L_t, D_t, H_t)$ satisfy the preceding hypotheses, with $L_s \ge L_t$ as in Remark 2.1. Using the arithmetic constant $r_{\mathbb{F}}$ from Appendix B.2, set*

$$C_0 := r_{\mathbb{F}} \left[L_s D_s |\mathbb{F}|^{D_{h,s}+1} + L_t D_t |\mathbb{F}|^{D_{h,t}+1} \right] + 1, \qquad C_1 := \binom{r_{\mathbb{F}} L_s + |\Sigma|}{|\Sigma|}. \tag{43}$$

For $C := \min\{C_0, C_1\}$, agreement on all words of length at most C implies agreement on every word:

$$T_s|_{\le C} \equiv T_t|_{\le C} \quad \Longrightarrow \quad T_s \equiv T_t.$$

Proof. Corollaries D.1 and D.2 give $\mathrm{ht}(\mathcal{A}_\Delta) \le \min\{C_0, C_1\} = C$. If $T_s \not\equiv T_t$, let w be a shortest word accepted by $\mathcal{A}_\Delta$. Its run contains no self-loop as deleting the corresponding letter would give a shorter accepted word. Every letter therefore induces a strict transition, so $|w| \le \mathrm{ht}(\mathcal{A}_\Delta) \le C$, contradicting agreement on $\Sigma^{\le C}$. □

D.4 Probabilistic length generalization

We now show that low error up to a sufficiently large cutoff persists at all larger lengths. The argument is automata-theoretic: a random input makes uniformly positive progress through a partially ordered DFA until it reaches an absorbing state.

Let $\mathcal{A} = (Q, \Sigma, \delta, q_0, B)$ be a partially ordered DFA, with bad states $B \subseteq Q$, and let $s = |\Sigma| \ge 2$. Let $C \ge 0$ be an integer upper bound on $\mathrm{ht}(\mathcal{A})$. For the difference function, take $\mathcal{A} = \mathcal{A}_\Delta$, with B its accepting states and $C = \min\{C_0, C_1\}$ from equation 43. For an exact length m, and for the distribution uniform over all words of length at most K, respectively, write

$$p_m \coloneqq \Pr_{w \sim \mathrm{Unif}(\Sigma^m)}[\delta(q_0, w) \in B], \qquad \mathcal{D}_K \coloneqq \mathrm{Unif}(\Sigma^{\le K}), \qquad e_K \coloneqq \Pr_{w \sim \mathcal{D}_K}[\delta(q_0, w) \in B].$$

Thus p_m is the bad-state probability at one fixed length, whereas e_K is the bad-state probability when every word of length at most K is equally likely. The mass that $\mathcal{D}_K$ assigns to length K is

$$\lambda_K \coloneqq \Pr_{w \sim \mathcal{D}_K}[|w| = K] = \frac{s^K}{1 + s + \cdots + s^K} = \frac{(s-1)s^K}{s^{K+1} - 1} > \frac{s-1}{s}.$$

Write Q_{abs} for the absorbing states of $\mathcal{A}$, namely those satisfying $\delta(q, a) = q$ for every $a \in \Sigma$. All other states are called transient.

Lemma D.5 (Absorption tail)**.** *Let* $0 < \epsilon < 1$. *If*

$$K \ge s\left(C + \log\frac{1}{\epsilon} + \sqrt{\log^2\frac{1}{\epsilon} + 2C\log\frac{1}{\epsilon}}\right), \tag{P}$$

then a uniformly random word $W_K \in \Sigma^K$ *ends in a transient state with probability at most* ϵ.

Proof. If $C = 0$, the initial state is absorbing and the claim is immediate. At every transient state, at least one letter makes a strict transition. Count a step as successful if it makes a strict transition or the run was already absorbed. Each step has conditional success probability at least $1/s$, so sequential coupling shows that the total success count stochastically dominates $Y \sim \mathrm{Bin}(K, 1/s)$. If the run is still transient after K steps, all successes were strict transitions and there were fewer than C of them. Hence

$$\Pr[\delta(q_0, W_K) \notin Q_{\mathrm{abs}}] \le \Pr[Y < C].$$

We use Chernoff's inequality to obtain an explicit sufficient cutoff.

Put $\mu = K/s$ and $\ell_\epsilon = \log(1/\epsilon)$. The displayed condition implies $\mu > C$. With $\eta = 1 - C/\mu \in (0, 1)$, the multiplicative Chernoff lower-tail bound[1] gives

$$\Pr[Y < C] \le \Pr[Y \le (1-\eta)\mu] \le \exp\left(-\frac{\mu\eta^2}{2}\right) = \exp\left(-\frac{(\mu - C)^2}{2\mu}\right).$$

It remains to make the exponent at least ℓ_ϵ. This is equivalent to

$$\mu^2 - 2(C + \ell_\epsilon)\mu + C^2 \ge 0.$$

The larger root of the left-hand quadratic is

$$C + \ell_\epsilon + \sqrt{\ell_\epsilon^2 + 2C\ell_\epsilon}.$$

Condition (P) is exactly the assertion that μ is at least this root. Hence $(\mu - C)^2/(2\mu) \ge \ell_\epsilon$, and the preceding display is at most $e^{-\ell_\epsilon} = \epsilon$. □

[1] See Wikipedia, "Chernoff bound: Multiplicative form (relative error)".

Theorem D.1 (Probabilistic length generalization)**.** *Let* $0 < \epsilon < 1$*. Suppose that* K *satisfies* (P) *and that*

$$\Pr_{w\sim\mathrm{Unif}(\Sigma^{\le K})}[\delta(q_0, w) \in B] \le \epsilon.$$

Then the probability of reaching a bad state remains small both at every fixed length $m \ge K$ *and over all words of length at most* $M \ge K$*:*

$$\Pr_{w\sim\mathrm{Unif}(\Sigma^{m})}[\delta(q_0, w) \in B] < 3\epsilon, \qquad \Pr_{w\sim\mathrm{Unif}(\Sigma^{\le M})}[\delta(q_0, w) \in B] < 3\epsilon.$$

Proof. The bad absorbing states at time K account for at most the bad probability conditioned on length K. Hence

$$\Pr[\delta(q_0, W_K) \in B \cap Q_{\mathrm{abs}}] \le p_K \le \frac{e_K}{\lambda_K} < \frac{s}{s-1}\epsilon \le 2\epsilon. \tag{A}$$

By Lemma D.5, the run is still transient at time K with probability at most ϵ. If a word of length $m > K$ ends in a bad state, then at time K its run was either transient or already in a bad absorbing state. Therefore, by (A),

$$\begin{aligned} p_m &\le p_K + \Pr[\delta(q_0, W_K) \notin Q_{\mathrm{abs}}] \\ &< \underbrace{2\epsilon}_{\text{from } e_K \le \epsilon} + \underbrace{\epsilon}_{\text{absorption tail}} = 3\epsilon \qquad (m > K). \end{aligned}$$

The same bound holds for $m = K$ by (A). Finally, e_M is a weighted average of e_K and the fixed-length errors $p_{K+1}, \dots, p_M$; each of these terms is strictly below 3ϵ. Thus $e_M < 3\epsilon$. □

Remark D.5. Keeping the alphabet dependence in the proof gives $p_m, e_M < (2 + 1/(s-1))\epsilon$. The upper-bound coefficient is 3 for binary alphabets and approaches 2 as s grows; this does not assert that the bound is attained.

D.5 Dependence of Chain Length on L and $\mathbb{F}$

We construct difference functions with unbounded poDFA height by varying either precision or depth, with the other architectural sizes fixed. Throughout, $\Sigma = \{a, b\}$, both letters have the same embedding, and outputs are scalar.

Implementation convention. Both constructions use one head and arithmetic satisfying Appendix B.2, with attention sums rounded after summation.

To select specific positions with attention, we operate the following way. Choose a negative score $z \in \mathbb{F}$ such that $\exp(z)$ rounds to zero. In particular, setting the convention that $\infty \cdot 0 = 0$, z can take the value $-\infty$. Selected positions then receive score 0 while unselected positions receive score z and value 0. Thus attention averages the selected values unless the denominator overflows.

Both constructions keep track of a fixed number of Boolean values, such as whether a position is selected. We represent each Boolean value using two coordinates: $(1, 0)$ for true and $(0, 1)$ for false. A separate coordinate stores the attention output.

To ensure layer normalization does not erase the information we keep two additional coordinates fixed at 0 and 1 before normalization. They allow the feed-forward map to recover the stored values and write them back as pairs $(1, 0), (0, 1)$.

D.5.1 Necessity of precision dependence

We create a family of framework where we increase precision while keeping both transformer architectures fixed. For each sufficiently large integer k, use the precision set

$$\mathbb{F}_k = \{0, \pm\infty\} \cup \{\pm j, \pm 1/j : 1 \le j \le k\} \cup \mathcal{C}_k.$$

Here $\mathcal{C}_k$ contains a bounded number of additional values needed for the attention scores and the Boolean encoding above, including its pointwise computations. Thus $p_k := |\mathbb{F}_k| = 4k + O(1)$ grows linearly with k. Choose the arithmetic so that positive attention sums exceeding k overflow to $+\infty$. Increasing k therefore allows uniform attention to average more positions before overflowing.

Proposition D.2 (Precision can force linear height growth)**.** *For the precision structures $\mathbb{F}_k$ above, there are pairs of transformers with fixed depths $L_s = L_t = 2$, fixed widths, and $H_s = H_t = 1$, whose reduced difference poDFAs satisfy*

$$\mathrm{ht}(\mathcal{A}_{\Delta_k}) = k + 2 = \Theta(p_k).$$

Thus the height can grow linearly with precision-set size while the architectures and alphabet remain fixed.

Proof. Both transformers first apply uniform attention with value 1 at every position. The resulting bit is

$$b(i) = \mathbf{1}\{2 \le i \le k + 1\}.$$

The teacher retains this bit through its second block. The student's second block attends only to earlier positions with $b(j) = 1$, using value $b(j)$, and takes the Boolean OR with the current bit. At most k positions are selected, so this attention never overflows when the selected set is nonempty. Consequently,

$$T_t(w)_i = \mathbf{1}\{2 \le i \le k + 1\}, \qquad T_s(w)_i = \mathbf{1}\{i \ge 2\}, \qquad \Delta_k(w) = \mathbf{1}\{|w| \ge k + 2\}.$$

The reduced DFA of $\{w : |w| \ge m\}$ counts length up to m, then remains in an accepting sink. Its $m + 1$ states are distinct: counts $r < t \le m$ are distinguished by a suffix of length $m - t$. Its height is therefore m, giving $k + 2$ here. Only Boolean attention outputs occur in this construction, so the encoding and feed-forward width are independent of k. Their pointwise computations and the score maps require only a bounded number of additional scalar values, as claimed for $\mathcal{C}_k$. □

Thus no height bound uniform over precision structures can ignore its precision set cardinality $|\mathbb{F}|$.

D.5.2 Necessity of linear depth dependence

Proposition D.3. *For one fixed precision structure $\mathbb{F}$, there are transformers T_m of depth $L = 2m$, with fixed width and one head, whose difference poDFAs against an always-zero transformer have height $m = L/2$.*

Proof. Fix a cutoff $k \ge 2$ and arithmetic implementing the preceding construction and the following register operations. Its two-layer student computes $\mathbf{1}\{i \ge 2\}$; complementing the last Boolean operation initializes a marker $u_1(i) = \mathbf{1}\{i = 1\}$.

Two further layers move a unique marker forward by one position. Given $u_j(i) = \mathbf{1}\{i = j\}$, the first attends only to the marker, with values u_j, and retains both u_j and

$$s_j(i) = u_j(i) \vee \mathbf{1}\{\text{an earlier marker exists}\} = \mathbf{1}\{i \ge j\}.$$

The second attends to positions selected by s_j, with values u_j. Its scalar output is

$$r_j(i) = \begin{cases} 1/(i-j), & 1 \le i - j \le k, \\ 0, & \text{otherwise.} \end{cases} \qquad u_{j+1}(i) = \mathbf{1}\{r_j(i) = 1\}.$$

Hence the new marker is exactly at $j + 1$. These layers manipulate a fixed number of registers and a fixed finite set of values, independently of j; the same precision structure and layer parameters suffice for every shift.

Initialization followed by $m - 1$ shifts uses $2m$ layers and yields

$$T_m(w)_i = \mathbf{1}\{i = m\}, \qquad \Delta_m(w) = \mathbf{1}\{|w| \ge m\}.$$

The preceding length-counter argument gives height m. □

With $\mathbb{F}$, Σ, D, and H fixed, this proposition proves the necessity in, at least, a linear dependence on depth in the bound on the height of a difference transformer.

E Data Complexity Bound

This appendix collects the statistical (in-distribution) generalization bound underlying Proposition 4.1. The core argument is imported from Rizvi-Martel et al. (2026), which studies the same student–teacher setting for single-output transformers; in order to keep the present paper self-contained, we restate the main result in our notation and sketch its proof (Appendix E.1), pointing the reader to Rizvi-Martel et al. (2026) for full details. We then describe the two modifications needed for our setting, namely students whose per-head dimension is smaller than the teacher's (Appendix E.2) and transformers producing an output at every position (Appendix E.3). Finally, we contrast the resulting G&C bound with the analogous bound for ERM (Appendix E.4).

Throughout, we write $D_{h,s} := D_s/H_s$ and $D_{h,t} := D_t/H_t$ for the per-head dimensions, and $D_{\mathrm{ff},s}, D_{\mathrm{ff},t}$ for the hidden widths of the feed-forward blocks. All parameters take values in the finite set $\mathbb{F}$, so that a transformer in $\mathcal{FPT}(L, D, H)$ has finitely many configurations and G&C samples uniformly among them.

E.1 Main statement and proof sketch

We say that a student T_s is *teacher-equivalent* (TE) to T_t on $\mathcal{D}_K$ if $\Pr_{w\sim\mathcal{D}_K}[T_s(w) \neq T_t(w)] = 0$. Let $\tilde{p}$ denote the probability that a uniformly random $T_s \in \mathcal{FPT}(L_s, D_s, H_s)$ is TE. The following is the general form of the G&C bound.

Theorem E.1 (Statistical generalization for G&C; adapted from Theorem B.1 and Corollary B.1 of Rizvi-Martel et al. (2026))**.** *Fix $\epsilon \in (0,1)$ and $\delta \in (0, \frac{1}{5})$. Let $T_t \in \mathcal{FPT}(L_t, D_t, H_t)$ be a teacher and $\mathcal{H} = \mathcal{FPT}(L_s, D_s, H_s)$ a hypothesis class with*

$$L_s \geq L_t, \qquad D_s \geq D_t, \qquad H_s \geq H_t, \qquad D_{h,s} \geq D_{h,t}, \qquad D_{\mathrm{ff},s} \geq D_{\mathrm{ff},t}.$$

Then

$$\tilde{p} \;\geq\; |\mathbb{F}|^{-B},$$

where

$$\begin{aligned} B \;:=\; & |\Sigma|(D_s - D_t) \\ & +\; L_t\Big(H_t\big(D_t D_{h,t} + 2D_t D_{h,s}\big) \\ & +\quad (H_s - H_t)\, D_t D_{h,s} + D_s D_t + D_s D_{\mathrm{ff},t} + D_t D_{\mathrm{ff},s} + D_s + D_{\mathrm{ff},s}\Big) \\ & +\; (L_s - L_t)\big(H_s D_{h,s} + D_{\mathrm{ff},s} + 1\big) D_t. \end{aligned}$$

Consequently, if T_s is obtained by the G&C learner on an i.i.d. sample S of N strings from $\mathcal{D}_K$ with

$$N \;\geq\; \frac{1}{\epsilon}\big(B \log|\mathbb{F}| + 3\log(2/\delta)\big),$$

then, with probability at least $1-\delta$ over S and the draw of T_s,

$$\Pr_{w\sim\mathcal{D}_K}\big[T_s(w) \neq T_t(w)\big] < \epsilon.$$

Under the standard conventions $D = HD_h$ and $D_{\mathrm{ff}} = D$, one has $B \sim \mathcal{O}\big(L_t D_t^2 + D_s D_t(2L_s + 3L_t)\big)$.

The condition $D_{h,s} \geq D_{h,t}$ is relaxed in Appendix E.2, and the readout term required for per-position outputs is added in Appendix E.3; Proposition 4.1 is the combination of Theorem E.1 with these two modifications and with the ERM bound of Appendix E.4.

Proof sketch. The proof has two ingredients.

From $\tilde{p}$ to a sample-complexity bound. The main lemma of Buzaglo et al. (2024b) states that, for any finite hypothesis class and any $0/1$ loss, if the probability of sampling a TE hypothesis under the prior is $\tilde{p} < \frac{1}{2}$, then G&C generalizes to error below ϵ with probability at least $1-\delta$ as soon as $N \geq (-\log\tilde{p} + 3\log(2/\delta))/\epsilon$. Hence it suffices to lower bound $\tilde{p}$, and $\tilde{p} \geq |\mathbb{F}|^{-B}$ yields the stated bound.

Lower bounding $\tilde{p}$ by constraint counting. We exhibit one explicit set of B parameter values such that any student agreeing with them on those B coordinates (and arbitrary elsewhere) computes exactly the function of T_t. Since parameters are drawn uniformly from $\mathbb{F}$, this event has probability exactly $|\mathbb{F}|^{-B}$, and $\tilde{p}$ is at least as large. The construction proceeds layer by layer and maintains the invariant that, at every position, the residual stream of the student has the form $[\lambda^\star, \mathbf{0}]$, where $\lambda^\star \in \mathbb{F}^{D_t}$ is the residual stream of the teacher and the remaining $D_s - D_t$ coordinates are zero.

- *Embedding.* The bottom $D_s - D_t$ coordinates of every token embedding are set to zero; this contributes $|\Sigma|(D_s - D_t)$ constraints.
- *Teacher layers ($\ell \le L_t$).* For each of the H_t teacher heads, the top-left $D_t \times D_{h,t}$ blocks of the student's $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$ are set to the teacher's matrices, and the blocks of $\mathbf{W}_Q, \mathbf{W}_V$ acting on the extra head coordinates are set to zero; since the query is zero on these coordinates, $\mathbf{W}_K$ may be left free there. For the remaining $H_s - H_t$ heads only the top block of $\mathbf{W}_V$ is zeroed, so that they contribute nothing to the teacher coordinates. The output projection and the feed-forward block are constrained analogously: their teacher blocks are copied and the blocks that would leak into (or read from) the extra coordinates are zeroed, including the biases. This preserves the invariant and contributes the $L_t(\cdot)$ term of B; see Lemma B.2 of Rizvi-Martel et al. (2026) for the block matrices.
- *Weak-identity layers ($L_t < \ell \le L_s$).* The extra $L_s - L_t$ layers need only preserve the top D_t coordinates, not the full residual. Setting the first D_t columns of the attention output projection, of the second feed-forward matrix, and of the corresponding bias to zero suffices; the bottom coordinates may then take arbitrary values without affecting the teacher block. This contributes $(H_s D_{h,s} + D_{\mathrm{ff},s} + 1)D_t$ constraints per layer, which is the last term of B (Theorem B.1 of Rizvi-Martel et al. (2026)).

Summing the three contributions gives B, and the asymptotic form follows by substituting $D = HD_h$ and $D_{\mathrm{ff}} = D$. □

E.2 Smaller student head dimension

Theorem E.1 requires $D_{h,s} \ge D_{h,t}$. In our experiments we vary H_s at fixed D_s, so that the student may have more but smaller heads than the teacher ($H_s \ge H_t$, $D_s \ge D_t$, but $D_{h,s} < D_{h,t}$). The construction extends to this case with only local modifications to the multi-head allocation, provided the larger teacher heads can be covered by several student heads; we therefore assume that $k := \lceil D_{h,t}/D_{h,s} \rceil$ satisfies $kH_t \le H_s$.

The direct block-embedding of the teacher's per-head matrices $\mathbf{W}_Q^\star, \mathbf{W}_K^\star, \mathbf{W}_V^\star \in \mathbb{F}^{D_t \times D_{h,t}}$ into a single student head of width $D_{h,s} < D_{h,t}$ no longer fits; we resolve this by allocating k student heads per teacher head, which is possible by the assumption $kH_t \le H_s$. Within each group of k student heads, we impose identical $\mathbf{W}_Q, \mathbf{W}_K$ blocks, so that all k heads share the same softmax attention pattern, and we distribute the teacher's $\mathbf{W}_V^\star$ (together with its slice of the output projection) across the k heads. The shared-pattern requirement asks that the teacher's $\mathbf{W}_Q^\star(\mathbf{W}_K^\star)^\top$ admit a rank-$D_{h,s}$ factorization; in the fixed-precision regime this is automatic whenever the training length satisfies $K \le D_{h,s}$, since every $K \times K$ attention matrix then has rank at most $D_{h,s}$. The parameter accounting is preserved asymptotically: the $H_t k$ student heads jointly contribute $\mathcal{O}(H_t k\, D_t\, D_{h,s}) = \mathcal{O}(D_t^2)$ constraints, and the remaining $H_s - H_t k$ heads add $\mathcal{O}((H_s - H_t k)\, D_t\, D_{h,s})$ constraints, so that the attention-side count remains $\mathcal{O}(D_s D_t)$ as in Theorem E.1.

E.3 Per-position outputs

Theorem E.1 is stated for a single output per string, as in Rizvi-Martel et al. (2026). transformers in $\mathcal{FPT}$ instead return a vector of dimension ν at every position through the pointwise affine output layer O (Definition 2.1). We write $\mathbf{U} \in \mathbb{F}^{\nu \times D_s}$ and $\mathbf{u} \in \mathbb{F}^\nu$ for the student's output weights and bias, and $\mathbf{U}^\star \in \mathbb{F}^{\nu \times D_t}$, $\mathbf{u}^\star \in \mathbb{F}^\nu$ for the teacher's. Teacher-equivalence is understood in the strict per-string sense: T_s is TE if, with probability one over $w \sim \mathcal{D}_K$, its output matches the teacher's at

every position. The corresponding empirical loss on a sample $(w^{(n)})_{n\le N}$ is

$$\frac{1}{N}\sum_{n=1}^{N}\mathbb{I}\Big[\exists\, i \le |w^{(n)}| \;:\; T_s(w^{(n)})_i \neq T_t(w^{(n)})_i\Big],$$

which is $\{0,1\}$-valued per string and therefore fits the G&C lemma of Buzaglo et al. (2024b) without change.

The constraint-counting argument carries over verbatim: the construction of Appendix E.1 preserves the teacher's residual stream in the top D_t coordinates at every position, not only at the last one. The only additional constraints come from the output layer. Writing the student's final residual stream at position i as $[\lambda_i^\star, \tilde{\lambda}_i]$, matching the teacher's output at every position requires the first D_t columns of $\mathbf{U}$ to equal $\mathbf{U}^\star$, the remaining $D_s - D_t$ columns to be zero (so that $\tilde{\lambda}_i$ does not leak into the output), and $\mathbf{u} = \mathbf{u}^\star$. This fixes every entry of the output layer, adding exactly $(D_s+1)\,\nu$ constraints, so that the constraint count becomes

$$B_{\text{seq}} \;=\; B \;+\; (D_s+1)\,\nu.$$

Since ν is fixed by the task, this term is subleading whenever $\nu \le D_t$, and the asymptotics of Theorem E.1 are unchanged.

E.4 Comparison to ERM

An analogous sample-complexity bound holds under Empirical Risk Minimization, obtained by a standard uniform-convergence argument over the finite hypothesis class $\mathcal{FPT}(L_s, D_s, H_s, \mathbb{F})$ (Appendix B.1 of Buzaglo et al. (2024b)). Counting parameters layer by layer under $D_s = H_s D_{h,s}$ and $D_{\text{ff},s} = D_s$ ($3H_sD_sD_{h,s} = 3D_s^2$ for the query, key and value matrices, D_s^2 for the output projection, $2D_s^2$ for the two feed-forward matrices, and $2D_s$ for the biases), the total parameter count of the student, and hence the ERM complexity term, is

$$B_{ERM} \;=\; L_s\big(6D_s^2 + 2D_s\big) + (D_s+1)\,\nu \;\sim\; \mathcal{O}\big(L_s D_s^2\big),$$

so that ERM generalizes to error below ϵ with probability at least $1-\delta$ whenever

$$N \;\ge\; \frac{1}{\epsilon}\big(B_{ERM}\log|\mathbb{F}| + \log(1/\delta)\big).$$

Contrasting this with $B \sim \mathcal{O}(L_sD_sD_t)$ from Theorem E.1, ERM scales with the *square* of the student width, whereas G&C scales with the *product* of the student and teacher widths; when the teacher is genuinely narrow ($D_t \ll D_s$), G&C therefore saves a factor of D_s/D_t over ERM. This narrow-teacher advantage is intrinsic to the volume-based argument underlying G&C, and is not captured by capacity-based bounds (Edelman et al., 2022; Trauger & Tewari, 2024; Yang et al., 2026c).

F Main theorem and Tradeoff

F.1 Main theorem

The main result given by Theorem 4.1 simply combines two results: sufficiently many samples give small error on $\mathcal{D}_K$, and a sufficiently large cutoff K extends that guarantee to every $\mathcal{D}_M$ with $M \geq K$.

Theorem F.1. *Fix $\epsilon \in (0,1)$, $\delta \in (0, \frac{1}{5})$, and $|\Sigma| \geq 2$. Let the student $T_s \in \mathcal{FPT}(L_s, D_s, H_s)$ and teacher $T_t \in \mathcal{FPT}(L_t, D_t, H_t)$ satisfy the learning framework of Remark 2.1, and the arithmetic assumptions of Appendix B.2. Let B be an admissible G&C complexity term from Proposition 4.1, including the output-layer constraints, and set $C = \min\{C_0, C_1\}$ as in equation 2.*

If T_s is given by the Guess & Check procedure, agreeing with T_t on N i.i.d. samples from $\mathcal{D}_K$, with

$$N \geq \frac{1}{\epsilon}\big(B \log |\mathbb{F}| + 3\log(2/\delta)\big) \tag{44}$$

and

$$K \geq |\Sigma| \left(C + \ln(1/\epsilon) + \sqrt{\ln^2(1/\epsilon) + 2C\ln(1/\epsilon)}\right), \tag{45}$$

then, with probability at least $1-\delta$ over the training sample and the G&C draw, we have

$$\forall M \geq K, \qquad \Pr_{\mathcal{D}_M}\left[T_s(w) \neq T_t(w)\right] \leq 3\epsilon. \tag{46}$$

Proof. By Proposition 4.1, the condition on N gives, with probability at least $1-\delta$,

$$e_K := \Pr_{w \sim \mathcal{D}_K}[T_s(w) \neq T_t(w)] < \epsilon.$$

Fix any student in this event. Its difference function's poDFA has height at most C by Corollaries D.1 and D.2. The condition on K is exactly the cutoff required by Theorem D.1, which gives $e_M < 3\epsilon$ for every $M \geq K$. This implication is deterministic once the student is fixed, so no additional failure probability or union bound over M is needed. □

Hidden dependencies. Under the feed-forward width convention used in Appendix E, a sufficient sample budget has order

$$\mathcal{O}\left(\frac{[L_t D_t^2 + D_s D_t(2L_s + 3L_t) + |\Sigma|D_s + (D_s+1)\nu]\log|\mathbb{F}| + \log(1/\delta)}{\epsilon}\right).$$

This retains the embedding and output-layer terms; the implicit constants are those of the data-complexity bound. Additionally, for $\ell = \ln(1/\epsilon)$, $\sqrt{\ell^2 + 2C\ell} \leq C + \ell$, so the sufficient length cutoff from Theorem D.1 is $\mathcal{O}(|\Sigma|(C + \ln(1/\epsilon)))$. Finally The first height bound C_0 contains $|\mathbb{F}|^{D_h} = \exp(D_h \ln|\mathbb{F}|)$ while the second bound C_1 is $\mathcal{O}(L_s^{|\Sigma|})$ when $\mathbb{F}$ and Σ are fixed and that $L_s \geq L_t$.

For ERM, the same proof applies after replacing the sample condition by the ERM condition in Proposition 4.1; the length cutoff is unchanged.

F.2 Multiplicative upper bound

We prove Theorem 4.2 from Section 4.3. For this, we importantly show that, whichever absorbing state the automaton eventually reaches, it has probability at least $a_C(K)$ of reaching that state within K steps. This lets us control eventual disagreement using the disagreement already visible at length K.

Theorem F.2 (Joint generalization bound in sample size and length)**.** *Fix $s = |\Sigma| \geq 2$, $\delta \in (0, \frac{1}{5})$, and integers $N \geq 1$ and $K \geq 0$. Work under the learning and arithmetic assumptions of Remark 2.1 and Appendix B.2. Let T_t be the teacher and let T_s be obtained by G&C from a labeled sample $S \sim \mathcal{D}_K^N$, where $\mathcal{D}_K = \mathrm{Unif}(\Sigma^{\leq K})$. Let B be an admissible G&C complexity term from Proposition 4.1, let $C = \min\{C_0, C_1\}$ be the uniform height bound in equation 43, and set*

$$a_C(K) := \Pr[\mathrm{Bin}(K, 1/s) \geq C], \qquad \rho := \frac{s}{s-1}, \qquad B_\delta := B\log|\mathbb{F}| + 3\log(2/\delta).$$

If $a_C(K) > 0$, then, with probability at least $1 - \delta$ over S and the G&C draw, for every integer $M \geq K$,

$$\Pr_{w \sim \mathcal{D}_M}[T_s(w) \neq T_t(w)] \leq \frac{\rho B_\delta}{N\, a_C(K)}. \tag{47}$$

For ERM, the same statement holds with an admissible ERM complexity term $B = B_{ERM}$ from Proposition 4.1; the probability is then only over S.

We first establish the bound on the absorption time of an infinite run of letters.

Lemma F.1. *Let $\mathcal{A}$ be a reduced poDFA of height at most C, driven by an infinite stream of i.i.d. uniform letters from Σ. Write Q_n for its state after n letters, T for its first hitting time of its absorbing state, and $Z = Q_T$ the absorbing state eventually reached. Then $T < \infty$ and for every absorbing state z,*

$$\Pr(T \leq K \mid Z = z) \geq a_C(K).$$

Proof. At each transient state, the probability of leaving is at least $1/s$, so absorption must occurs within C strict transitions. A self-loop leaves the state unchanged, and future letters are independent; thus conditioning on $Z = z$ does not change the self-loop probability. Even under this conditioning, each step before absorption therefore makes progress with probability at least $1/s$. By coupling with independent Bernoulli$(1/s)$ trials, absorption is no slower than obtaining C successes, giving

$$\Pr(T \leq K \mid Z = z) \geq \Pr[\mathrm{Bin}(K, 1/s) \geq C] = a_C(K).$$

□

Proof of the theorem. For a fixed student, write

$$p_m := \Pr_{w \sim \mathrm{Unif}(\Sigma^{=m})}[T_s(w) \neq T_t(w)], \qquad e_m := \Pr_{w \sim \mathcal{D}_m}[T_s(w) \neq T_t(w)].$$

We have by convention that disagreement means a mismatch at any output position, so the difference poDFA $\mathcal{A}_\Delta$ has at most a unique absorbing accepting state $\perp$, with its height bounded by C.

Run $\mathcal{A}_\Delta$ on an i.i.d. uniform stream, with T, Z as in the lemma, and set $q_\infty := \Pr(Z = \perp)$ the probability that the ultimate absorbing state is the disagreement one. Since $\perp$ is absorbing,

$$p_K = \Pr(T \leq K,\ Z = \perp) = q_\infty \Pr(T \leq K \mid Z = \perp) \geq a_C(K) q_\infty.$$

Moreover, persistence of disagreement makes p_m nondecreasing, so for every M,

$$e_M = \frac{\sum_{m=0}^{M} s^m p_m}{\sum_{m=0}^{M} s^m} \leq \frac{\sum_{m=0}^{M} s^m p_M}{\sum_{m=0}^{M} s^m} = p_M \leq q_\infty.$$

To connect the exact-length quantity p_K to the training distribution, observe that

$$\lambda_K := \Pr_{w \sim \mathcal{D}_K}(|w| = K) = \frac{s^K}{\sum_{m=0}^{K} s^m} = \frac{(s-1)s^K}{s^{K+1} - 1} > \frac{s-1}{s} = \rho^{-1}.$$

Therefore $e_K \geq \lambda_K p_K \geq \lambda_K a_C(K) q_\infty$, which implies

$$e_M \leq q_\infty \leq \frac{e_K}{\lambda_K a_C(K)} \leq \frac{\rho e_K}{a_C(K)}. \tag{48}$$

Finally, Proposition 4.1 gives $e_K \leq B_\delta / N$ with probability at least $1 - \delta$, as we can apply the proposition with $\epsilon = B_\delta / N$. Substituting into equation 48 proves the result simultaneously for every $M \geq K$. The ERM version is obtained identically, thanks to $\log(1/\delta) \leq 3 \log(2/\delta)$. □

The factor $a_C(K)$ measures how much of the eventual disagreement is already visible at length K and the factor ρ accounts for drawing training words uniformly from all words of length at most K.

F.3 Tradeoff between Length and Data Complexity

We give the calculations behind the tradeoff discussed in Section 4.3 and its implications for the allocation of a given token-budget. Keep the notation of Theorem F.2, with $s = |\Sigma| \geq 2$, $C \geq 1$, and $\rho = s/(s-1)$. Fix a target error $\epsilon \in (0,1)$ and confidence $1-\delta$. The theorem gives the sufficient sample size

$$N_{\mathrm{req}}(K) := \frac{N_0}{a_C(K)}, \qquad N_0 := \frac{\rho B_\delta}{\epsilon}, \qquad K \geq C. \tag{49}$$

Here N_0 is the limit sample size as $K \to \infty$, that would still satisfy the target error ϵ. Halving the target error doubles the N_0's and so it does for $N_{\mathrm{req}}(K)$, at every length, without changing its dependence on K.

Three length regimes. The following table summarizes the sample sizes certified by the bound and already discussed in Section 4.3.

Proposition F.1 (Sample size as a function of length)**.**

training length	*sufficient number of samples*
$K < C$	*no finite N is certified*
$K = C$	$N_{\mathrm{req}}(C) = s^C N_0$
$K \geq sC$	$N_0 \leq N_{\mathrm{req}}(K) \leq 2N_0$

(50)

Thus increasing length from C to sC reduces the sufficient sample size by a factor of at least $s^C/2$.

Proof. For $K < C$, $a_C(K) = 0$, while $a_C(C) = s^{-C}$. At $K = sC$, the binomial has integer mean C, which is also a median (Kaas & Buhrman, 1980, Theorem 1), so $a_C(sC) \geq 1/2$. The last row follows because $a_C(K)$ increases to one. □

How to invest a fixed token budget. Write $\mathrm{Budget} = NK$ for the token budget, an upper bound on the total number of tokens when sampling from $\mathcal{D}_K$. We keep K integral and ignore rounding of N. Substituting $N = \mathrm{Budget}/K$ into the theorem gives

$$\Pr_{w \sim \mathcal{D}_M}[T_s(w) \neq T_t(w)] \leq \frac{\rho B_\delta}{\mathrm{Budget}} \frac{K}{a_C(K)} \qquad (M \geq K), \tag{51}$$

with probability at least $1-\delta$. Thus the best allocation must maximize $a_C(K)/K$.

Corollary F.1 (Optimal token allocation)**.** *Suppose $\mathrm{Budget} \geq sC$. The smallest value of the upper bound in equation 51 is*

$$\epsilon^\star(\mathrm{Budget}) = \frac{\rho B_\delta}{\mathrm{Budget}} \min_{C \leq K \leq \mathrm{Budget}} \frac{K}{a_C(K)},$$

attained by

$$K^* \in \arg\max_{C \leq K \leq \mathrm{Budget}} \frac{a_C(K)}{K}, \qquad N^* = \frac{\mathrm{Budget}}{K^*}.$$

Moreover,

$$\frac{\rho B_\delta sC}{\mathrm{Budget}} \leq \epsilon^\star(\mathrm{Budget}) \leq \frac{2\rho B_\delta sC}{\mathrm{Budget}}. \tag{52}$$

In particular, choosing $K = sC$ gives an error bound within a factor two of the optimum.

Proof. Only $K/a_C(K)$ depends on the allocation. For $Y \sim \mathrm{Bin}(K, 1/s)$, Markov's inequality gives

$$a_C(K) = \Pr(Y \geq C) \leq \frac{\mathbb{E}[Y]}{C} = \frac{K}{sC}, \qquad \text{hence} \qquad \frac{K}{a_C(K)} \geq sC.$$

Conversely, $a_C(sC) \geq 1/2$, so choosing $K = sC$ gives $K/a_C(K) \leq 2sC$. □

There is no closed form formula to find the optimum, but the exact optimizer can be found by checking only $C \leq K \leq \min\{\mathrm{Budget}, 2sC\}$: every $K > 2sC$ has $K/a_C(K) \geq K > 2sC$ and cannot improve on $K = sC$. At $K = C$, the same ratio is Cs^C. For $C \geq 2$, choosing $K = sC$ therefore improves the error bound at a fixed token budget by a factor of at least $s^{C-1}/2$.

An explicit choice of K. For $C \geq 2$, define

$$\widehat{K} := \left\lceil s\left(C + \log C + \sqrt{\log^2 C + 2C\log C}\right)\right\rceil. \tag{53}$$

The Chernoff calculation in the proof of Lemma D.5, with tail probability $1/C$, gives $a_C(\widehat{K}) \geq 1 - 1/C$. If the budget allows $\widehat{K} \leq \text{Budget}$, let $\widehat{\epsilon}(\text{Budget})$ be the bound obtained with $K = \widehat{K}$ and $N = \text{Budget}\,/\widehat{K}$. Then

$$1 \leq \frac{\widehat{\epsilon}(\text{Budget})}{\epsilon^\star(\text{Budget})} \leq \frac{\widehat{K}}{sC(1-1/C)} = 1 + \mathcal{O}\left(\sqrt{\frac{\log C}{C}}\right).$$

Hence $\widehat{K} = sC + \mathcal{O}(s\sqrt{C\log C})$ is asymptotically optimal as $C \to \infty$. The simpler choice $K = sC$ gives the uniform factor-two guarantee above. This justify the choice of the bound on K in the joint statistical and length generalization main result given by Theorem 4.1.

Least token budget for a target error. The same calculation can be read in reverse. By equation 49, the least token budget certified for error ϵ is

$$\text{Budget}_{\min}(\epsilon) = N_0 \min_{K \geq C} \frac{K}{a_C(K)}, \qquad N_0 sC \leq \text{Budget}_{\min}(\epsilon) \leq 2N_0 sC.$$

Choose a minimizing length K and take $N = N_0/a_C(K)$. Thus both problems—minimizing error at a fixed budget and minimising budget at a fixed error—reduce to the same ratio $K/a_C(K)$ optimization.

F.4 Adaptive main theorem: uniform training length

We adapt Theorem 4.1 to the distribution that first chooses a length uniformly, then draws a word of that length:

$$\widetilde{\mathcal{D}}_K: \quad k \sim \text{Unif}\{0,\ldots,K\}, \qquad w \sim \text{Unif}(\Sigma^k).$$

The sample-size condition is unchanged. For the length transfer, we use the fact that at least half of the sampled lengths are at least $K/2$.

Corollary F.2 (Main theorem for uniform training lengths)**.** *Fix $\epsilon \in (0, \frac{1}{2})$, $\delta \in (0, \frac{1}{5})$, and $s = |\Sigma| \geq 2$. Use the learning and arithmetic assumptions of Theorem F.1, with the same complexity term B and height bound C. Let T_s be obtained by G&C from a labeled sample $S \sim \widetilde{\mathcal{D}}_K^N$. If*

$$N \geq \frac{B\log|\mathbb{F}| + 3\log(2/\delta)}{\epsilon}$$

and

$$\left\lceil \frac{K}{2} \right\rceil \geq s\left(C + \log\frac{1}{2\epsilon} + \sqrt{\log^2\frac{1}{2\epsilon} + 2C\log\frac{1}{2\epsilon}}\right),$$

then, with probability at least $1-\delta$ over S and the G&C draw,

$$\forall M \geq K, \qquad \Pr_{w\sim\widetilde{\mathcal{D}}_M}[T_s(w) \neq T_t(w)] \leq 4\epsilon.$$

Proof. The statistical argument of Proposition 4.1 applies to any input distribution, so the condition on N gives error at most ϵ on $\widetilde{\mathcal{D}}_K$ with probability at least $1-\delta$. Fix a student in this event and write

$$p_m := \Pr_{w\sim\text{Unif}(\Sigma^m)}[T_s(w) \neq T_t(w)], \qquad \widetilde{e}_m := \frac{1}{m+1}\sum_{k=0}^{m} p_k.$$

Since $I = \{\lceil K/2\rceil, \ldots, K\}$ contains at least half the lengths,

$$\frac{1}{|I|}\sum_{k\in I} p_k \leq \frac{K+1}{|I|}\widetilde{e}_K \leq 2\epsilon.$$

Hence some $\widetilde{K} \in I$ satisfies $p_{\widetilde{K}} \leq 2\epsilon$. The difference poDFA has height at most C. By the condition on K and Lemma D.5, it is still transient after $\widetilde{K}$ letters with probability at most 2ϵ. Any disagreement at a later length either already occurred by $\widetilde{K}$, or the run was still transient then. Thus $p_m \leq 4\epsilon$ for every $m > \widetilde{K}$. Averaging gives, for every $M \geq K$,

$$\widetilde{e}_M \leq \frac{K+1}{M+1}\epsilon + \frac{M-K}{M+1}\,4\epsilon \leq 4\epsilon.$$

This holds for all M on the same statistical event. □

As usual, for ERM, use its sample-size condition from Proposition 4.1. Here, the see that the bound given on K is roughly two times larger. However the rates do not change, and as in Theorem 4.1, the sufficient length remains $\mathcal{O}(s(C + \log(1/\epsilon)))$.

Remark F.1. The multiplicative tradeoff bound from Theorem 4.2 also extends to $\widetilde{\mathcal{D}}_K$. At least half of its mass lies on lengths between $\lceil K/2 \rceil$ and K. Thus, the same averaging argument finds a length in this interval with error at most twice the average error. Repeating the proof of Theorem F.2 therefore gives a bound of order

$$\frac{2B_\delta}{N\, a_C(\lceil K/2 \rceil)}$$

on every later error. The factor 2 replaces the factor ρ as it represents the mass of the upper half of the length range.

G Extensions

G.1 Local Attention

We keep fixed precision and strict causal masking. Each head is either global or local; a local head at position i attends only to $\max(1, i-W), \ldots, i-1$, for a common window size $W \geq 1$. Let T_s, T_t have depths L_s, L_t and the same output dimension. Set

$$s = |\Sigma| \geq 2, \qquad R = W \max(L_s, L_t).$$

We consider disagreement at ordinary token positions,

$$\Delta(w) = \mathbf{1}\{\exists i \leq |w| : \; T_s(w)_i \neq T_t(w)_i\}.$$

By causality, $\Delta(u) = 1$ implies $\Delta(uv) = 1$. We do not include disagreement evaluated only at EOS.

Definition G.1 (Cache–poDFA cascade)**.** *Let* $\mathrm{suffix}_R(u)$ *retain the last* $\min(R, |u|)$ *symbols of* u*. A cache–poDFA cascade is a DFA with states* $(c, q) \in \Sigma^{\leq R} \times Q$*, initial state* (ϵ, q_0)*, and transitions*

$$\delta((c, q), a) = \big(\mathrm{suffix}_R(ca), f(q, c, a)\big), \qquad q \preceq f(q, c, a), \tag{54}$$

where $(Q, \preceq)$ *is a finite partially ordered set. Its accepting states are those with* $b(q) = 1$*, for a flag* $b : Q \to \{0, 1\}$ *satisfying* $b(q) \leq b(f(q, c, a))$*. The ordered component is a poDFA over* $\Sigma^{\leq R} \times \Sigma$*: it reads the cache together with the next letter. Let* $C_{\mathrm{loc}} \geq 1$ *be an integer bound on the number of strict changes of* q *along a run, including a change of the disagreement flag.*

Proposition G.1 (Representation of disagreement)**.** *The language* $\{w : \Delta(w) = 1\}$ *is recognized by a cache–poDFA cascade of radius* R*. In particular, the full DFA need not be partially ordered.*

Proof. Write $Y^j\varphi$ to mean that φ held j positions earlier, with value false if that position does not exist. The finite-precision compilation of Li & Cotterell (2026) gives a pointed formula $\psi_\Delta \in \mathrm{LTL}[\mathbf{P}, Y]$ for disagreement at the current position. A local head uses the current representation and the previous W representations, adding shifts Y^j with $j \leq W$. Global heads introduce only Boolean operations and $\mathbf{P}$, as in Appendix C.3. Thus the total shift along any formula branch is at most R.

Move every shift inward using

$$Y\mathbf{P}\varphi \equiv \mathbf{P}Y\varphi, \qquad Y(\varphi \wedge \theta) \equiv Y\varphi \wedge Y\theta, \qquad Y\neg\varphi \equiv Y\top \wedge \neg Y\varphi.$$

The resulting formula, still denoted ψ_Δ, uses local atoms $Y^j\pi_a, Y^j\top$ with $0 \leq j \leq R$, Boolean operations, and $\mathbf{P}$. The cache c and current letter a determine all local atoms; $|c|$ also records which earlier positions exist.

Suppose t symbols have been read, and the next symbol is $a = w_{t+1}$. For each past subformula $\chi = \mathbf{P}\theta$, store

$$z_\chi = \mathbf{1}\{\exists j \leq t : \; w, j \models \theta\}.$$

This is the truth of χ before reading a. The cache is $c = \mathrm{suffix}_R(w_1 \cdots w_t)$.

For any normalized subformula φ, define $\mathrm{val}_\varphi(z, c, a)$ as its truth value at the current position. With $m = |c|$ and positions in ca numbered from 1 to $m+1$, compute it recursively as follows:

$$\begin{aligned}
\mathrm{val}_{Y^j\pi_d}(z, c, a) &= \begin{cases} \mathbf{1}\{(ca)_{m+1-j} = d\}, & j \leq m, \\ 0, & j > m, \end{cases} \\
\mathrm{val}_{Y^j\top}(z, c, a) &= \mathbf{1}\{j \leq m\}, \\
\mathrm{val}_{\mathbf{P}\eta}(z, c, a) &= z_{\mathbf{P}\eta}, \\
\mathrm{val}_{\neg\varphi}(z, c, a) &= 1 - \mathrm{val}_\varphi(z, c, a), \\
\mathrm{val}_{\varphi\wedge\eta}(z, c, a) &= \mathrm{val}_\varphi(z, c, a) \wedge \mathrm{val}_\eta(z, c, a).
\end{aligned}$$

Here $d \in \Sigma$, $0 \leq j \leq R$, and Y^0 is the identity.

The state update is

$$z'_\chi = z_\chi \vee \mathrm{val}_\theta(z, c, a), \qquad b' = b \vee \mathrm{val}_{\psi_\Delta}(z, c, a).$$

These updates record whether θ, or a disagreement, has occurred. Both use the old profile z, so past operators exclude the current position.

Starting from $z = b = 0$, induction gives the stated meaning of z and $b = \Delta(w_1 \cdots w_t)$. Each bit changes only from 0 to 1, so $q = (z, b)$ is partially ordered. Let G count the distinct past subformulas before moving the shifts inward. Each gives at most $R+1$ shifted versions; adding the disagreement flag gives

$$C_{\mathrm{loc}} = (R + 1)G + 1. \tag{55}$$

□

Remark G.1 (Scope of the height bound)**.** This construction extends the subformula-count bound of Appendix D.1, but does not extend the temporal-profile bound of Appendix D.2. Each shifted past test changes from false to true at most once, so counting these tests bounds the ordered component's height. Temporal profiles over Σ do not record the recent suffix needed by local attention. Adding the cache changes the alphabet, so the original profile bound no longer applies.

Definition G.2 (Settled states and absorption)**.** *A reachable state (c, q) is* settled *if either $b(q) = 1$ (disagreement has occurred), or every continuation keeps $b = 0$ (disagreement is impossible). Otherwise it is* transient*. Merge the first kind into a bad sink and the second into a safe sink. Both sets are closed under transitions, so this preserves the language. The merged states are absorbing; let τ be the first time the run reaches either of them.*

Both transformers share the same suffix cache, which depends only on the input. The cache may keep changing after the outcome is settled. If the ordered component can never advance, the state is settled.

Lemma G.1 (Progress in a fresh block)**.** *From every transient state, some word of length at most $B = R + 1$ strictly advances the ordered component. Consequently, for independent uniform input symbols,*

$$\Pr[\textit{strict transition in the poDFA in the next } B \textit{ steps} \mid \textit{history}] \geq \alpha, \qquad \alpha = |\Sigma|^{-B}. \tag{56}$$

Proof. A transient state admits a continuation causing disagreement. Take a word v ending at the first strict advance of the ordered component. If $|v| \leq B$, use v. Otherwise use its final B symbols. Either this shorter word advances the ordered component within R steps, or those steps restore the cache just before the last symbol of v, while q stays fixed. The last symbol then causes the same advance. This word has length $r \leq B$ and occurs at the start of the next block with probability $s^{-r} \geq s^{-B}$, given any history. □

Theorem G.1 (Absorption tail with local attention)**.** *For an i.i.d. uniform input stream, every integer $K \geq 0$ satisfies*

$$\Pr[\tau > K] \leq \Pr\left[\mathrm{Bin}\left(\left\lfloor \frac{K}{B} \right\rfloor, \alpha\right) < C_{\mathrm{loc}}\right], \qquad B = R + 1, \quad \alpha = s^{-B}. \tag{57}$$

In particular, for $0 < \epsilon < 1$, write $\ell_\epsilon = \log(1/\epsilon)$. The right-hand side is at most ϵ whenever

$$K \geq B\left\lceil s^B \left(C_{\mathrm{loc}} + \ell_\epsilon + \sqrt{\ell_\epsilon^2 + 2C_{\mathrm{loc}}\ell_\epsilon}\right)\right\rceil. \tag{58}$$

This gives the sufficient scale $K = O\big((R + 1)s^{R+1}(C_{\mathrm{loc}} + \log(1/\epsilon))\big)$.

Proof. Split the stream into blocks of length B. Count a block as successful if it contains an ordered advance or a settled state. By Lemma G.1, its success probability is at least α, given all previous blocks. Coupling with independent Bernoulli(α) trials shows that the number of successes in n blocks stochastically dominates $\mathrm{Bin}(n, \alpha)$.

If $\tau > nB$, there are fewer than C_{loc} successful blocks: each contains a strict advance, and C_{loc} advances would leave the state settled. Taking $n = \lfloor K/B \rfloor$ proves Equation 57. For $\mu = \lfloor K/B \rfloor \alpha > C_{\mathrm{loc}}$, the Chernoff bound gives

$$\Pr[\mathrm{Bin}(\lfloor K/B \rfloor, \alpha) < C_{\mathrm{loc}}] \leq \exp\left(-\frac{(\mu - C_{\mathrm{loc}})^2}{2\mu}\right).$$

Condition equation 58 makes this at most ϵ. □

Corollary G.1 (Length generalization)**.** *Let* $p_k = \Pr_{w\sim\mathrm{Unif}(\Sigma^k)}[\Delta(w) = 1]$ *and* $e_K = \Pr_{w\sim\mathrm{Unif}(\Sigma^{\le K})}[\Delta(w) = 1]$. *For every* $m \ge K$,

$$p_m \le p_K + \Pr[\tau > K].$$

Thus, if K *satisfies equation 58, then* $p_K \le \epsilon$ *implies* $p_m \le 2\epsilon$ *for all* $m \ge K$. *Likewise,* $e_K \le \epsilon$ *implies* $p_m < 3\epsilon$ *and* $e_M < 3\epsilon$ *for all* $m, M \ge K$.

Proof. Couple different lengths as prefixes of one infinite stream. If a disagreement occurs after time K but not by time K, the state at time K must be transient. This proves the first inequality. For bounded-length sampling,

$$p_K \le e_K/\lambda_K, \qquad \lambda_K = \frac{s^K}{1 + s + \cdots + s^K} > \frac{s-1}{s} \ge \frac{1}{2}.$$

The tail bound gives $p_m < 3\epsilon$; averaging the errors at lengths above K together with e_K gives the same bound for e_M. □

A uniform bound on C_{loc} lets us combine this corollary with statistical generalization, keeping the same confidence. Local attention replaces single-step progress by blocks of length $R + 1$, with progress probability at least $s^{-(R+1)}$.

G.2 Generalization with Nonuniform Distribution over Vocabulary

Our main length generalization bound is stated for a uniform distribution over the vocabulary Σ for simplicity. In fact, the proof yields a generalization to a nonuniform distribution P over Σ. In fact, even dependencies between successive symbols are allowed; in this case we can consider distributions P over semi-infinite strings $X \in \Sigma^{\omega}$[2]. We only need to assume that every poDFA over Σ reaches an absorbing state in finite time almost surely on $X \sim P$; this is satisfied, e.g., if for each $i > 0$ and each $\sigma \in \Sigma$:

$$P(X_i = \sigma | X_{1\ldots i-1} = \xi) \ge \epsilon_P \tag{59}$$

for every $\xi \in \Sigma^{i-1}$, for some $\epsilon_P > 0$.

Definition G.3. *Let* $\mathcal{A}$ *be a poDFA, and* P *be a distribution over semi-infinite strings in* Σ^{ω}. *Define the following random variables over semi-infinite strings* $X \sim P$*:*

1. $T \in \mathbb{Z}$*: time of reaching an absorbing state*
2. Z*: the absorbing state eventually reached*

Then define

$$a_{\mathcal{A}}(K) := \min_{z\in Q_{absorb}} \Pr[T \le K | Z = z] \tag{60}$$

where Q_{absorb} *denotes the set of absorbing states of* $\mathcal{A}$*; we restrict to states where* $\Pr[Z = z] > 0$.

To keep the same length distribution as $\mathcal{D}_K$, set $s = |\Sigma| \ge 2$ and define $\mathcal{D}_K^P$ by drawing $X \sim P$ and an independent length L with

$$\Pr(L = \ell) = \frac{s^\ell}{\sum_{j=0}^K s^j}, \qquad 0 \le \ell \le K, \qquad w = X_{1:L}.$$

When P is the i.i.d. uniform source, $\mathcal{D}_K^P = \mathcal{D}_K$.

Then the proof of our joint generalization bound (Theorem 4.2) carries through for $\mathcal{D}_K^P$, replacing the quantity $a_C(K)$ used in that theorem with a quantity defined from $a_{\mathcal{A}}(K)$:

Theorem G.2 (Joint Generalization Bound in Sample Size and Length, Version of Theorem 4.2 for General Input Distribution)**.** *Use the same hypotheses as Theorem F.2, with teacher* T_t *and hypothesis class* $\mathcal{H}$*, and define*

$$a(K) := \inf_{T_s\in\mathcal{H}} a_{\mathcal{A}(\Delta(T_t,T_s))}(K) \tag{61}$$

[2] I.e., X has the form $X_1X_2X_3\ldots$.

where $\mathcal{A}(\Delta(T_t, T_s))$ represents the difference automaton for T_t and T_s.[3]

Let T_s be trained on N i.i.d. samples from $\mathcal{D}_K^P$, with $a(K) > 0$. Then, for every $M \geq K$, the error rate on $\mathcal{D}_M^P$ satisfies

$$\Pr_{w \sim \mathcal{D}_M^P}[T_s(w) \neq T_t(w)] \leq \frac{\rho B_\delta}{N \cdot a(K)}, \tag{62}$$

with probability at least $1 - \delta$, where $\rho := \frac{s}{s-1}$ and $B_\delta := B \log |\mathbb{F}| + 3 \log(2/\delta)$.

The statement above assumes G&C; as usual, it also holds for ERM with $B = B_{ERM}$ from Proposition 4.1.

Proof. For a fixed student, write $e_m = \Pr_{w \sim \mathcal{D}_m^P}[T_s(w) \neq T_t(w)]$, and let p_K be the error probability on the exact prefix $X_{1:K}$. Let q be the probability of ever disagreeing on $X \sim P$. Since disagreement is absorbing, the definition of $a(K)$ gives $p_K \geq a(K)q$, while $e_M \leq q$ for every M. The mass at length K is

$$\lambda_K = \frac{s^K}{\sum_{j=0}^K s^j} > \frac{s-1}{s} = \rho^{-1}.$$

Thus $e_K \geq \lambda_K p_K \geq \lambda_K a(K) q$, and

$$e_M \leq q \leq \frac{e_K}{\lambda_K a(K)} \leq \frac{\rho e_K}{a(K)}.$$

The statistical argument of Proposition 4.1 applies to $\mathcal{D}_K^P$ and gives $e_K \leq B_\delta / N$ with probability at least $1 - \delta$ (trivially if $B_\delta / N \geq 1$). Substitution proves the claim, with the same ERM adaptation as in Theorem F.2. □

G.3 Lower Bound on Tradeoff

We show a *lower bound* also exhibiting a *multiplicative interaction* between N and $a(K)$. Our lower bound on the tradeoff matches the more general version allowing nonuniform input distributions (Appendix G.2).

Theorem G.3. *Let $C, d \in \mathbb{N}$. There is a family $\mathcal{F}_{C,d}$ of target functions f representable by fixed-precision transformers, such that the following holds. For every $0 < \epsilon \leq \frac{1}{4}$ there is an input distribution P such that, the following holds:*

Consider a learner with hypothesis class $\mathcal{H} := \mathcal{F}_{C,d}$ producing a hypothesis $T_s \in \mathcal{H}$ from N i.i.d. samples at length K (each sample the length-K prefix of a sample from P, labeled with the output provided by the target f).

Then, define the risk $R_{N,K}$ by averaging the test loss on very long test strings:

$$\lim \sup_{K_{test} \to \infty} \Pr_{w \sim P}(T_s(w|_{1 \ldots K_{test}}) \neq f(w|_{1 \ldots K_{test}}))$$

across targets $f \in \mathcal{F}_{C,d}$ and size-N i.i.d. datasets.

Then, for all N and all $K \geq C + 2$

$$R_{N,K} \leq \epsilon \quad \Rightarrow \quad \epsilon \geq \frac{\log_2 |\mathcal{F}_{C,d}|}{8 N a(K)}$$

where $a(K)$ is as defined in Eq. 61.

Note here that, in the setting of the function classes $\mathcal{F}_{C,d}$ built here, this bound closely resembles Theorem G.2: the numerator is the negative log probability that a uniform sample from $\mathcal{H}$ matches any given ground truth f – which is essentially the statistical complexity term B arising in the G&C bound (which is the negative log probability that a random sample from the hypothesis class

[3] As in Corollary D.1 and Proposition D.1.

matches the target, as per discussion in Appendix E) – whereas the denominator is proportional to the product of N and $a(K)$, as in Theorem G.2. Thus, any bound applying at the generality of (i) general hypothesis classes $\mathcal{H} \subseteq \mathcal{FPT}(L, D, H)$, (ii) general distributions P over Σ^ω needs to have essentially this form, as we have both upper and lower bounds on ϵ of this form up to proportionality factors.

Proof. We define P over Σ^ω so that successive symbols are always sampled independently; we thus specify the probability at the level of individual symbols (i.e., specify P as a probability over Σ). Define

$$\Sigma := \{0, 1\} \times [d] \times \{0, 1\} \tag{63}$$

with elements (g, j, p), with

$$P((g, j, p)) = P_1(g)P_2(j)P_3(p) \tag{64}$$

where

$$P_1(g = 1) = \eta \qquad P_3(p = 1) = r$$

and P_2 is uniform.

We parameterize the family by (i) the parameter $d \in \mathbb{N}$, (ii) the parameter $C \in \mathbb{N}$, (iii) a bit vector $v \in \{0, 1\}^d$, and (iv) a tilt $\eta > 0$, (v) a tilt $r \in (0, 1)$. Here, (i)–(iii) jointly define f; (iv) – (v) define the distribution P. We later choose η based on ϵ, as $\eta := 4\epsilon$.

Given a target $v = (v_1, \ldots, v_d) \in \{0, 1\}^d$, we first define a poDFA as follows. The state set is

$$\mathcal{Q} := \{q_0, \bot\} \cup \{q_{j,i} : 0 \le i \le C; 1 \le j \le d\}. \tag{65}$$

1. Start in q_0.
2. Consider the first token, of the form (g, j, p)
   (a) If $g \neq 1$, go to $\bot$.
   (b) Else, j indexes a chain to go into. That is, go to $q_{j,0}$.
3. Once we are in such a state

$$\delta(q_{j,i}, (g, j', p)) := \begin{cases} q_{j,i+1} & \text{if } i < C, p = 1 \\ q_{j,i} & \text{else} \end{cases} \tag{66}$$

   That is, each branch is $q_{j,0} \to q_{j,1} \to \cdots \to q_{j,C}$ where only $p = 1$ advances the chain

   Also, set $\delta(\bot, \sigma) \equiv \bot$ for every $\sigma \in \Sigma$.
4. Every state q_0, $\bot$, and $q_{j,i}$ with $i < C$ outputs 0. Endpoint $q_{j,C}$ outputs an indicator for v_j; that is for every $\sigma \in \Sigma$,

$$\mathfrak{o}(q_{j,C}, \sigma) = v_j \tag{67}$$

This is represented by a family of FPTs with $L = \Theta(C)$, $D = d$, $H = 1$.

Now for any two distinct elements of $\mathcal{F}_{C,d}$, we note that the difference automaton $\mathcal{A}_\Delta$ has a very similar structure: it consists of chains of length C again, and the symbols have the same effect of advancing in a chain as they have in the poDFAs given above.

Define

$$a_C(t) := \Pr\left[\mathrm{Bin}(t, r) \ge C\right] \tag{68}$$

This in fact matches the definition of $a(K)$ from Eq. 61.

A single example reveals v_j with probability

$$q_j(K) = \frac{\eta a_C(K - 2)}{d} \tag{69}$$

After N independent examples, coordinate j remains "unrevealed" with prob

$$\left(1 - \frac{\eta a_C(K - 2)}{d}\right)^N \tag{70}$$

Consider test length m. For a given coordinate j, the chance of it becoming relevant on a test sample here is

$$\frac{\eta a_C(m-2)}{d} \tag{71}$$

The test error rate is:

$$\text{TestErrorRate} \geq \frac{\eta a_C(m-2)}{2}\left(1-\frac{\eta a_C(K-2)}{d}\right)^N \tag{72}$$

Setting $m \to \infty$:

$$\text{TestErrorRate} \geq \frac{\eta}{2}\left(1-\frac{\eta a_C(K-2)}{d}\right)^N \tag{73}$$

Take $\eta = 4\epsilon$:

$$\text{TestErrorRate} \geq 2\epsilon\left(1-\frac{4\epsilon a_C(K-2)}{d}\right)^N \tag{74}$$

If TestErrorRate $\leq \epsilon$, then, dividing both sides by ϵ, and applying Bernoulli's Inequality

$$\frac{1}{2} \geq \left(1-\frac{4\epsilon a_C(K-2)}{d}\right)^N \geq 1-\frac{4N\epsilon a_C(K-2)}{d}$$

Hence,

$$\frac{4N\epsilon a_C(K-2)}{d} \geq \frac{1}{2}$$

Rearranging, and by monotonicity of a_C,

$$\epsilon \geq \frac{d}{8N a_C(K)} \tag{75}$$

Now, by construction, $d = \log_2 |\mathcal{F}_{C,d}|$. This concludes the result. □

H EXPERIMENTS

We use three controlled synthetic task families to test the qualitative predictions of our theory. We choose these families for variety in teacher depth, width, number of heads, alphabet size, and associated automaton height. The induction-head task also tests local positional information provided by a local attention head. The experiments naturally reflect the fixed-precision setting: all computations use finite-precision arithmetic (FP32 in our implementation). First, the logarithmic–linear tradeoff between training length K and sample size N suggests that an additive increase in K can offset a multiplicative reduction in N: increasing training length can therefore be more effective than adding examples. Second, larger student depth L, width D, or alphabet size $|\Sigma|$ can increase the sufficient requirements on N and/or K, suggesting possible performance degradation at fixed training resources. These experiments also have an exploratory role: the two bounds on the automaton height C in Lemma 3.1 yield different dependencies on these quantities, and worst-case guarantees need not describe their empirical effects.

H.1 TASK FAMILIES

H.1.1 L_k: ALTERNATING LANGUAGES

Let the alphabet be $\Sigma_L = \{a, b\}$. The language L_k consists of the words that begin with a and that contain at least k alternating constant blocks of a's or b's. Equivalently, writing the unique block decomposition of a nonempty word as $w = r_1^+ \cdots r_m^+$, with $r_i^+ = r_i r_i^*$ and where the consecutive $r_i, r_{i+1} \in \{a, b\}$ are distinct.

$$L_k := \left\{ w = r_1^+ \cdots r_m^+ \in \{a, b\}^* : r_1 = a \text{ and } m \geq k \right\}. \tag{76}$$

Thus L_3 accepts `aaabbbbbba` and `ababa`, but rejects `aaaabb` and `bbbaabb`. These tasks are inspired by the alternating-block depth hierarchy of Yang et al. (2025a); here we ask for at least k blocks rather than exactly k. We use this family to study increasing sequential depth, and hence the dependence on teacher depth L_t, while keeping the alphabet fixed.

Specifics. The residual stream gives access to the current token, which is included when counting blocks. The target at position i is 1 exactly when the prefix $w_1 \cdots w_i$ belongs to L_k; the EOS target is the membership label of the full word. Evaluation includes every ordinary position and EOS. Each new letter differs from the previous one with probability 0.05, rather than 0.5, and words start with a with probability 0.95. This makes alternations rarer and avoids early target saturation. The distribution still fits our framework: equivalently, one can use an alphabet of 20 symbols in which only one symbol causes a turn. Additionally, Appendix G.2 discusses the non-uniform extension for the the sample–length tradeoff multiplicative bound in Equation 10.

Example. For L_3, the target becomes 1 when the third block begins:

w	a	a	a	b	b	b	a	a	a	EOS
y	0	0	0	0	0	0	1	1	1	1

H.1.2 A_k: ALPHABET LANGUAGES

Here $\Sigma_k = \{s_1, \ldots, s_k\}$. These Alphabet languages ask whether every letter has occurred at least once in the word:

$$A_k := \{ w \in \Sigma_k^* : \forall j \in [k],\ \exists i \in [|w|] \text{ such that } w_i = s_j \}. \tag{77}$$

For example, A_5 accepts precisely the words over $\{a, b, c, d, e\}$ in which no letter is missing. We assume a reference teacher construction, where one attention head monitors each letter, giving $H_t = k$. With this hypothesis, increasing k therefore varies H_t, the corresponding residual width D_t, and $|\Sigma_k|$ together, rather than isolating head count.

Remark. Unlike the Alternating languages, we do not prove that representing A_k requires k teacher heads. We nevertheless use $H_t = k$ as a heuristic, supported by two main reasons. First, we empirically observed that learning the Alternating languages with less heads than alphabet symbols was difficult. Secondly, under a sufficiently skewed distribution, if one head monitors two letters

a, b, its attention mechanism may fail to detect a sporadic presence of letter b among a word full of a's: indeed, most of the attention mass would be set on a's, while fixed-precision rounding and saturation would potentially leave b undetected.

Specifics. Letters are sampled independently and uniformly. Each ordinary position is labelled by the membership of its prefix in A_k, and EOS by the membership of the full word. Evaluation includes all these positions.

Example. For A_3 over $\{a, b, c\}$, the target becomes 1 once every letter has appeared:

w	a	b	c	a	b	c	EOS
y	0	0	1	1	1	1	1

H.1.3 IH_k: Induction Head tasks

The Induction Head tasks have a vector-valued prediction at every ordinary token position; EOS is not supervised or evaluated. Here again, $\Sigma_k = \{s_1, \ldots, s_k\}$. For each $i \in [|w|]$, the target output is a vector $IH_k(w)_i \in \{0,1\}^k$, with one dimension per letter in Σ_k, whose d-th coordinate records whether the bigram $\langle w_i, s_d\rangle$ has appeared strictly earlier in the word:

$$\big(IH_k(w)_i\big)_d := \mathbf{1}\{\exists j \in \{1, \ldots, i-2\} : (w_j, w_{j+1}) = (w_i, s_d)\}, \qquad d \in [k]. \tag{78}$$

For instance, at the third position of `abac`, the current letter is a and the earlier bigram $\langle a, b\rangle$ has occurred, so the b-coordinate is 1. Thus IH_k is the product of k binary classification tasks, one for each possible successor s_d. Distinct letters must have distinct embeddings, so fixed precision requires $|\mathbb{F}|^{D_t} \geq k$. This family therefore varies the alphabet size and the teacher's width requirement, as well as the output dimension k; it does not isolate the effect of D_t.

Specifics. The induction-head task uses local positional information, unlike the preceding two families. A range-one local attention head can attend only to position $i-1$ at position $i > 1$, providing the information needed to form bigrams without a separate positional encoding. Appendix G.1 treats this extension through a finite suffix cache coupled to a partially ordered component. We use only its past-subformula-counting bound, corresponding to C_0 in Lemma 3.1, with the additional cache and block factors. The temporal-profile bound C_1 is not established for this extension.

Example. For IH_3 over $\{a, b, c\}$, write target coordinates in the order (a, b, c):

w	a	b	a	c	a	b
y	$(0,0,0)$	$(0,0,0)$	$(0,1,0)$	$(0,0,0)$	$(0,1,1)$	$(1,0,0)$

H.2 Protocol and evaluation

Binary loss and accuracy. Exact equality of FP32 outputs is not a useful experimental criterion, so we treat each target coordinate as a binary class. For a word w, let $\mathcal{S}(w)$ be its supervised positions, let $y_{i,d} \in \{0,1\}$ be the dth target coordinate at position i, and let $z_{i,d}$ be the corresponding model logit. We minimize

$$\mathcal{L}(\theta) = \frac{1}{\sum_{n=1}^{N} |\mathcal{S}(w^{(n)})|} \sum_{n=1}^{N} \sum_{i \in \mathcal{S}(w^{(n)})} \frac{1}{\nu} \sum_{d=1}^{\nu} \ell_{\mathrm{BCE}}\big(z_{i,d}^{(n)}, y_{i,d}^{(n)}\big), \tag{79}$$

$$\ell_{\mathrm{BCE}}(z, y) = -y \log \sigma(z) - (1-y) \log\big(1 - \sigma(z)\big), \tag{80}$$

with $\sigma(z) := (1 + e^{-z})^{-1}$. Note that the output dimension ν is only 1 for L_k and A_k, and $\nu = |\Sigma_k|$ for IH_k. Thus the first two tasks use one BCE term at every ordinary position and EOS, whereas IH_k averages BCE across all $|\Sigma_k|$ target coordinates at each ordinary position.

We classify a coordinate as $\widehat{y}_{i,d} = \mathbf{1}\{z_{i,d} \geq 0\}$ and a word is correct if $\widehat{y}_{i,d} = y_{i,d}$ for every $i \in \mathcal{S}(w)$ and $d \in [\nu]$. Accuracy is the fraction of correct words, and perfect accuracy means that every word is correct.

Remark. In the fixed-precision abstraction, the binary classifications given above, based on the logit regions $1\{z_{i,d} < 0\}$ and $\{z_{i,d} \geq 0\}$, are representable by boolean predicates in the same way as done in Proposition D.1. Equivalently, the binary classification framework naturally stems from an alternative definition of the difference function 29, checking on two logit regions rather than on exact equality.

Sampling and supervision. Unless stated otherwise, all ordinary input symbols are drawn independently and uniformly from the alphabet. Each sampled word has length exactly K, measured in ordinary symbols. Agreement requires an exact match at every supervised position, and the current letter can contribute through the residual stream.

For L_k and A_k, we additionally append and supervise an EOS token. These are language-recognition tasks, so the terminal readout of Li & Cotterell (2025) is natural, alongside our prefix-wise outputs. EOS is an experimental convention, not an assumption of our base framework. Requiring a correct EOS output adds a success condition, making evaluation at least as demanding as checking ordinary positions alone; we do not claim a measured effect on training difficulty. The persistent difference function in our bounds concerns ordinary positions, while EOS accuracy is an additional check.

Optimisation and repetitions. In all cases, optimisation uses Adam Kingma & Ba (2014) with learning rate searched $\in \{$2e-5, 5e-5, 1e-4$\}$, choosing the learning rate that induces the most seeds to fit perfectly the training data. Training lasts for at most $2{,}000$ epochs and stops early when the epoch training loss is below 1e-5. Unless the quantity N is explicitly given, the training sets contains $N = 256$ i.i.d. samples. All configurations in all the experimenets were run 10 times over 10 different seeds. Finally, a model is selected in our analyses if and only if it achieves perfect accuracy on its training set. This matches the consistency requirement of Guess-and-Check, but not its distribution over hypotheses.

Choice of training length. Unless K is swept, it is chosen using task-specific coverage criteria: approximately 80% of L_k words starting with a contain the required blocks, 80% of A_k words contain every letter, and 50% of IH_k words have exhibited every bigram before the final ordinary position. These criteria concern the target task. They were also partially selected empirically to make the experiments interesting (neither trivial nor impossible). The lower threshold for IH_k reflects the useful coordinate-wise supervision available before all bigrams have occurred. The A_k and IH_k choices are guided by coupon-collector estimates; overlapping bigrams are not independent draws.

Default student architectures. The trained transformers follow the structure of Appendix B.2, with the local-attention extension for IH_k. Outputs are evaluated by coordinate-wise binary classification rather than exact equality of the logits. Computations use PyTorch's default **FP32** precision. This reflects the fixed-precision setting, but does not by itself enforce the idealized summation conventions in Appendix B.2. The ReLU feed-forward layers have one hidden layer of width $2D_s$. Unless an experiment explicitly varies an architectural parameter, the trained student uses the following family-specific default architectures. Single-task sweeps vary one entry at a time; teacher comparisons instead use a common student backbone across task sizes, as described below.

- For L_k, the student has $L_s = k - 1$ layers, embedding dimension $D_s = 256$, and one attention head $H_s = 1$.
- For A_k, it has one layer $L_s = 1$ and $H_s = k$ attention heads. The global embedding dimension is fixed to $D_s = 360$.
- For IH_k, it has two layers $L_s = 2$ and embedding dimension $D_s = 256$, with two heads per layer: one local head of range 1 and one global head.

In the end, the evaluated models were selected to strike a balance between representational complexity and tractability, ensuring they were sufficiently expressive to yield meaningful insights while remaining learnable.

Evaluation. We evaluate at absolute test lengths, such as $K_{\text{test}} = 500$, or relative lengths, such as $K_{\text{test}} = 5K$. These lengths count ordinary symbols, excluding EOS. Each model is evaluated on $2{,}000$ independent test words of the specified length. The supervised positions are the ordinary

positions and EOS for L_k and A_k, and the ordinary positions only for IH_k; accuracy is as defined above.

To mirror the selection step of Guess-and-Check, as noticed above, we retain for plotting only models with perfect training accuracy. Consequently, a reported point can aggregate fewer than the 10 attempted seeds; a blank region in a plot means that no seed achieved perfect training accuracy for that configuration.

H.3 RESULTS

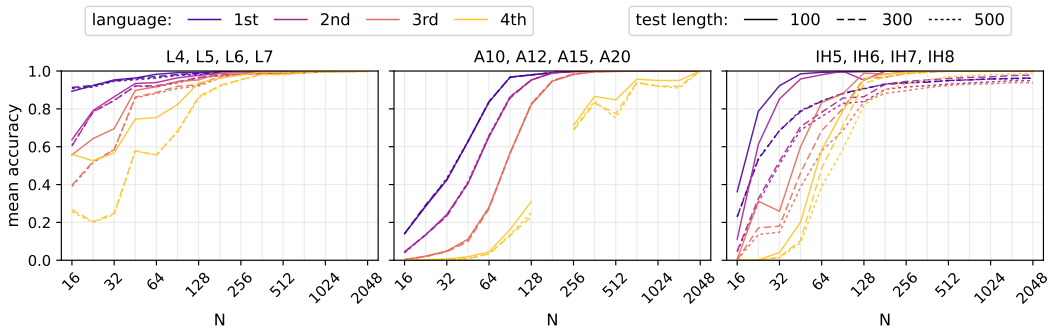
We first examine whether increasing the teacher's parameter size shifts the amount of data or training length needed for high test accuracy. In this section focusing on the impact of the teacher's size, for each multi-panel teacher comparison, the student backbone (L_s, D_s, H_s) is held fixed across tasks; input embeddings and output dimensions still follow the task alphabet and targets. Moreover, among each families, all students have the same parameter sizes. This is because we want to ideally study the effect of the teachers' sizes, excluding the variations of the students' sizes. To ensure realizability, we therefore set all students within the a same family adopt the structure of the largest teacher. For example, the experiments comparing the Alternating languages L_4, L_5, L_6, L_7, theory predicts that each student requires $k-1$ layers: we therefore set all the students to have $7-1=6$ layers. Similarly, Alphabet language students have all 20 heads, and the Induction Head tasks stay with there common embedding dimension of 256.

Teacher sizes refer to reference realizations of the target functions, whose labels are generated directly. The families probe different, sometimes coupled resources: depth L_t for L_k; head count H_t, width D_t, and alphabet size for A_k; and width requirements, alphabet size, and output dimension for IH_k.

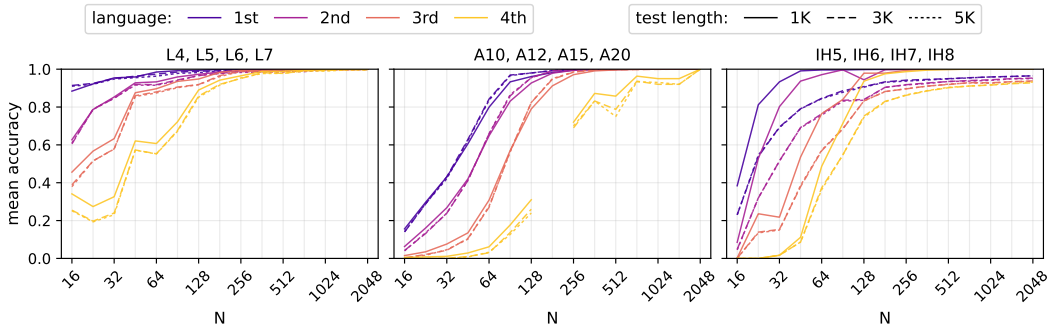
H.3.1 IMPACT OF THE TEACHER'S SIZE

The two height bounds in Lemma 3.1 give different dependencies: C_0 grows with depth and width, but decreases with head count at fixed width (other quantities held fixed); C_1 depends on depth and alphabet size, not width or head count. For global attention, the effective height bound is their minimum. A larger alphabet also lowers the worst-case probability of making progress in the automaton. These effects can increase the sufficient requirements on N and K, but do not predict monotone empirical performance. For IH_k, only the local extension of the C_0 argument is used.

Required data complexity. We first vary N and compare task sizes within each family. The student backbone is fixed, while K is adjusted using the coverage criteria above. Larger tasks therefore receive longer, potentially more informative examples, which could partly offset their increased difficulty. Nevertheless, the observed sample requirement generally increases with task size. For A_k and IH_k, this comparison does not separate teacher size from alphabet size or, for IH_k, the number of output coordinates that must all be correct.



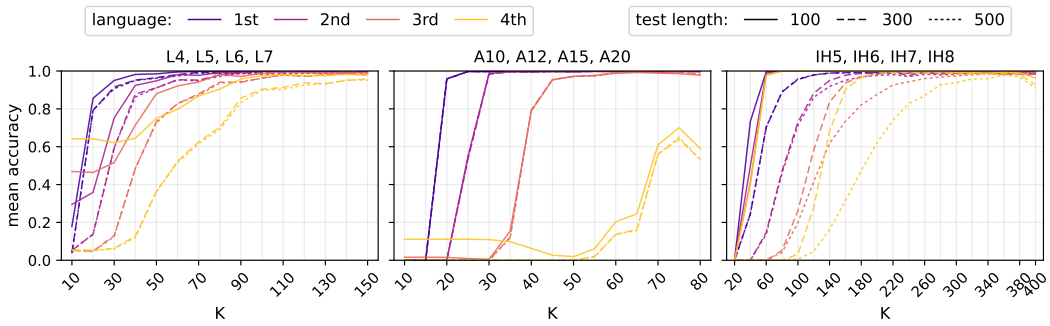
(a) Evaluation on absolute lengths



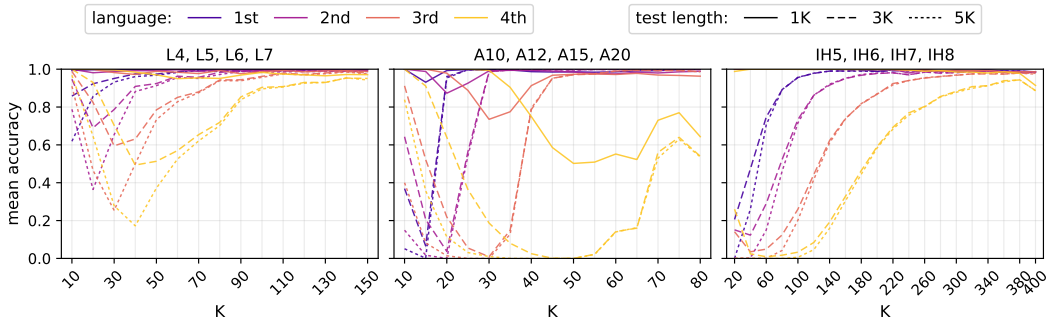
(b) Evaluation on relative lengths

Figure 6: **Impact of teacher's size on data complexity**. L_k probes teacher depth L_t; A_k jointly varies H_t, D_t, and $|\Sigma|$; IH_k varies alphabet size, width requirements, and output dimension. Student backbones are fixed within each family, while K follows the task-specific coverage criterion.

Required training length. We similarly sweep K, comparing task sizes at each common training length, with N and the student backbone fixed. Larger tasks generally require longer training words for high test accuracy. We observe that this is qualitatively consistent with the increased sufficient length scales.



(a) Evaluation on absolute lengths



(b) Evaluation on relative lengths

Figure 7: **Impact of teacher's size on length complexity**. L_k probes teacher depth L_t; A_k jointly varies H_t, D_t, and $|\Sigma|$; IH_k varies alphabet size, width requirements, and output dimension. Tasks are compared at common values of K, with N and student backbones fixed.

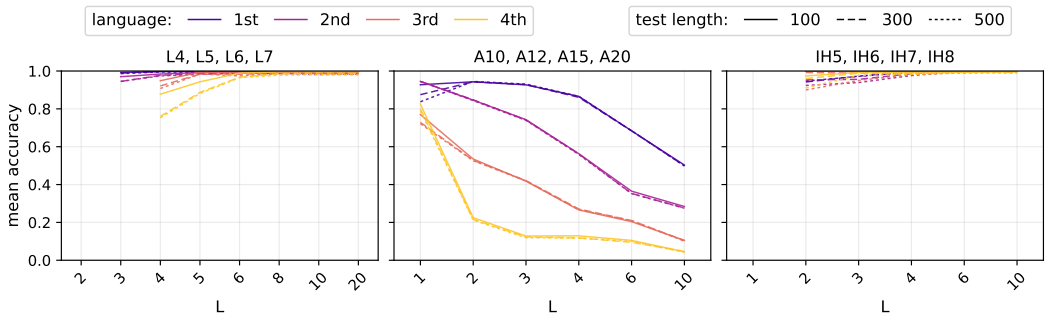
H.3.2 Impact of the student's size

We next vary the size of the student, L_s, D_s and H_s, while holding the teacher and all other independent experimental parameters fixed. Overall, the effect is not consistently monotone. But the bounds describe worst-case sufficient resources and added capacity can also improve learning. This distinction is relevant in overparameterized models, as discussed in Appendix A.

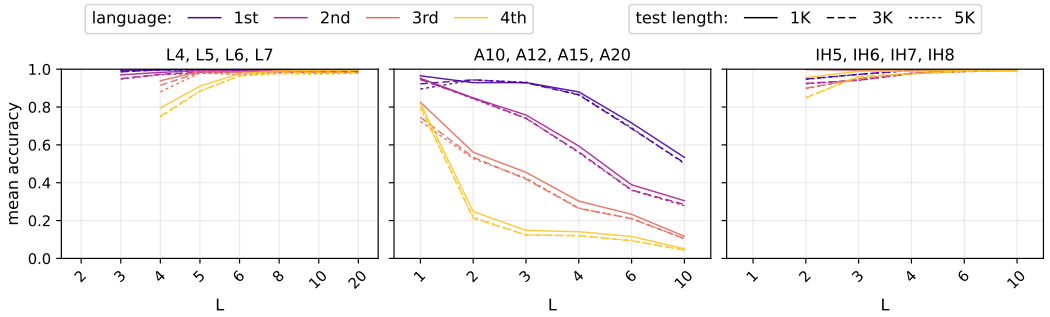
Student depth. Increasing the number of student layers L_s sometimes improves learning and length generalization. For A_k, however, it substantially degrades length generalization. We observe that is consistent with the depth dependence of both height bounds and the resulting sufficient requirements on N and K.

Student embedding dimension. Varying the student embedding dimension D_s has little visible effect on length generalization and, in some settings, improves learning. Thus, increasing D_s does not appear to impose a systematic data or length-complexity cost in these experiments. At fixed H_s, C_0 increases with D_s, whereas C_1 is independent of width. The observations therefore do not support a simple monotone width penalty; they do not identify which bound is informative. For IH_k, the comparison remains with the local extension of C_0, not C_1.

Student head count. Increasing the number of student attention heads H_s appears to slightly harm learning in the L_k and IH_k families, but substantially improves it in the A_k family, including beyond the k heads used in the reference construction. Fewer than k heads need not imply non-representability. At fixed D_s, increasing H_s reduces $D_{h,s} = D_s/H_s$ and hence the student contribution $r_{\mathbb{F}} L_s D_s |\mathbb{F}|^{D_s/H_s+1}$ to C_0; C_1 is unchanged. The improvement for A_k is notably compatible with this direction. For IH_k, the sweep also changes the numbers of local and global heads while keeping their proportions fixed.

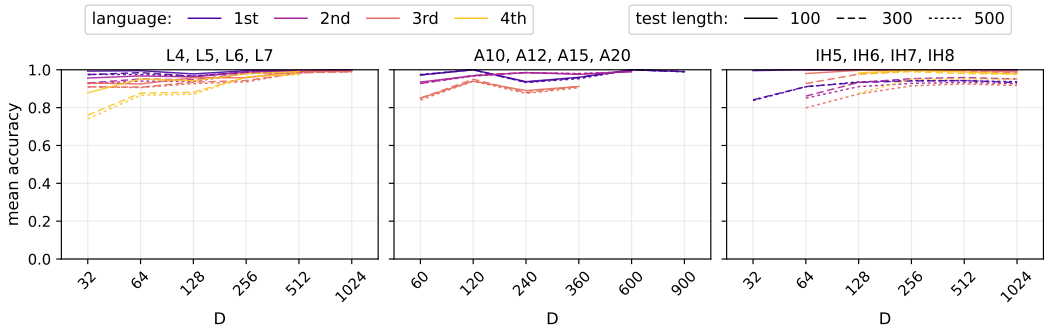


(a) Evaluation on absolute lengths

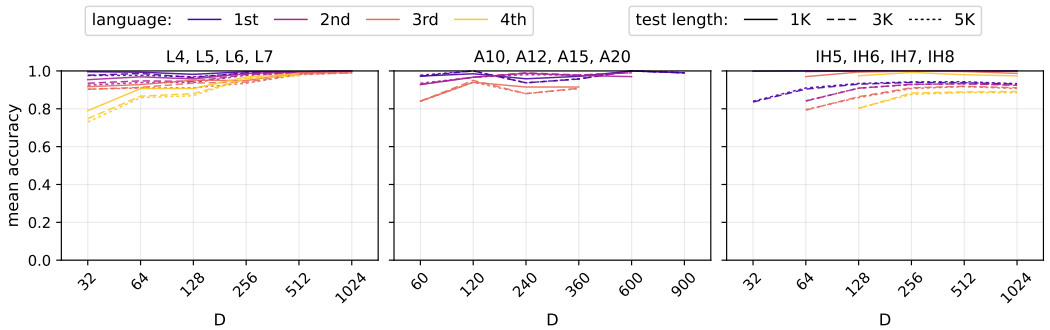


(b) Evaluation on relative lengths

Figure 8: **Impact of the student's depth on length generalization**. We vary L_s while holding D_s and H_s fixed.

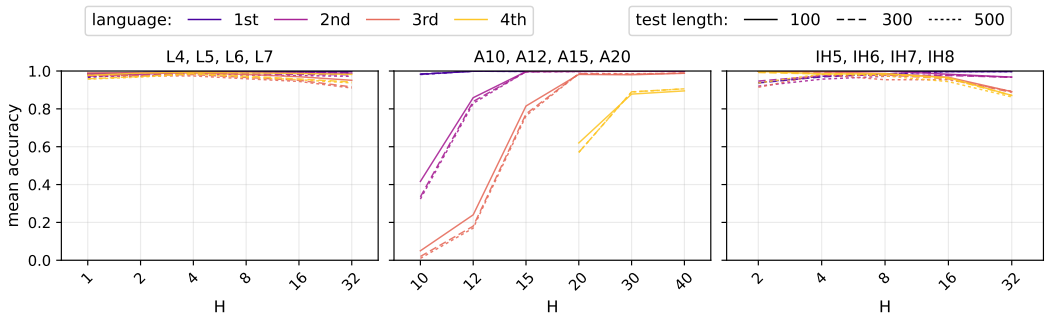


(a) Evaluation on absolute lengths

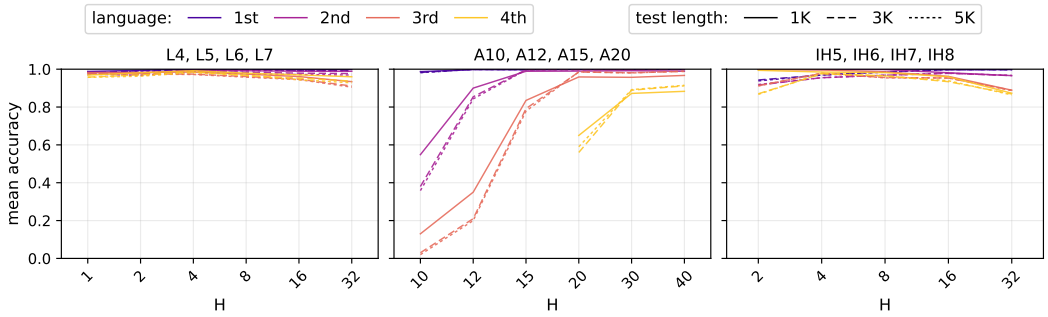


(b) Evaluation on relative lengths

Figure 9: **Impact of the student's embedding dimension on length generalization**. We vary D_s while holding L_s and H_s fixed.



(a) Evaluation on absolute lengths



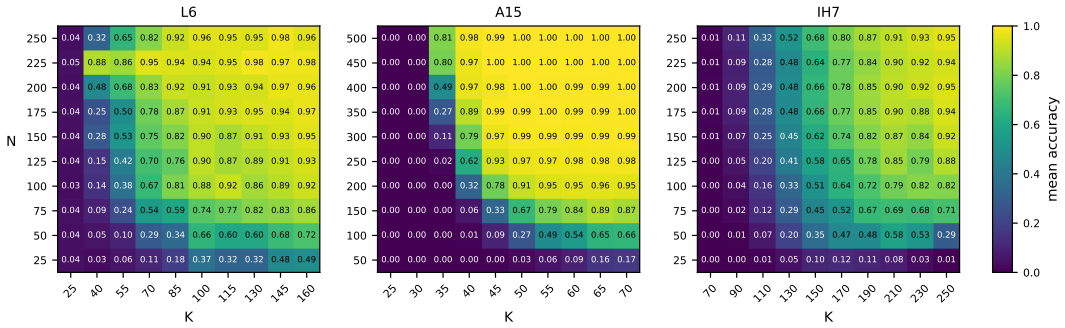
(b) Evaluation on relative lengths

Figure 10: **Impact of the student's head count on length generalization**. We vary H_s while holding L_s and D_s fixed.

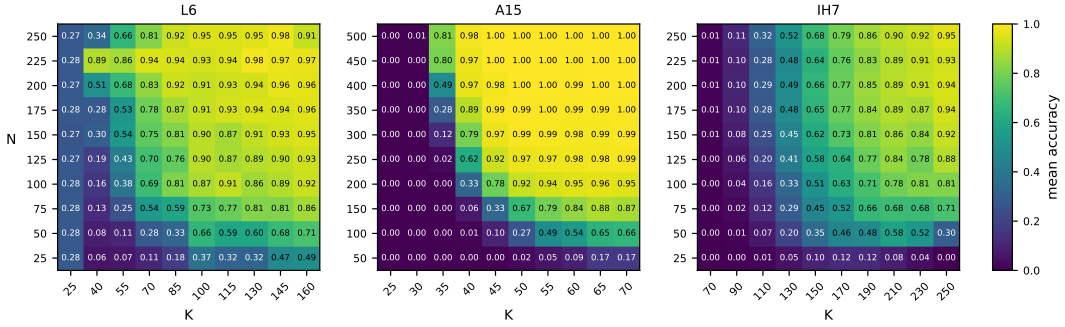
H.3.3 Tradeoff between training length and data complexity

For these tradeoff experiments, student models are set with the usual parameter sizes described in this section. In particular, L_6 has exactly 5 layers. However, we set the number of heads H_s to 20 – not 15– for A_{15}. The reason being that the Alphabet languages A_k were overall difficult to be trained on for students with exactly $H_s = k$. As our theoretical framework welcomes and over-parametrized student transformer, we simply decided to increase H_s to 20.

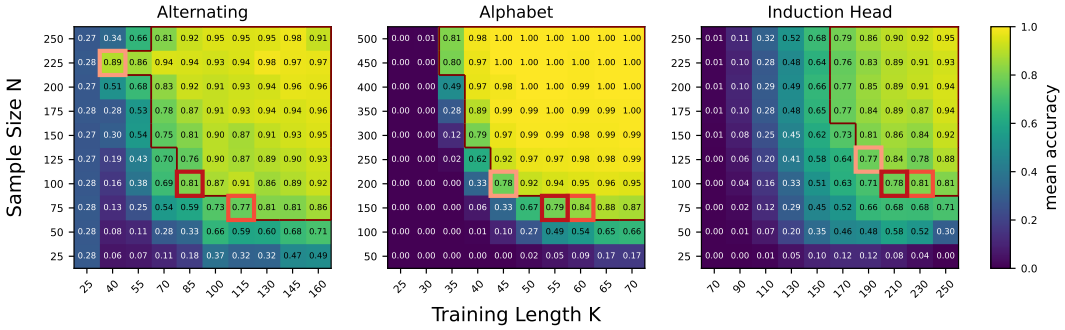
We observe a tradeoff between training length K and sample size N: longer examples can substantially reduce the number needed for length generalization. Both axes use linear scales. The observed frontier is qualitatively consistent with the logarithmic–linear tradeoff illustrated in Figure 3, but of course, these plots do not establish an exponential rate. For a fixed accuracy threshold, the configurations that achieve the threshold with the smallest token budget typically use a larger K and a smaller N. Conversely, for a fixed token budget, the configurations with the best accuracy likewise favor a larger K over a much larger N. This resembles the allocation suggested by Corollary F.1. For IH_k, the relevant theoretical comparison uses the blockwise local-attention extension of the C_0 argument. The plotted budgets NK count ordinary tokens; L_k and A_k additionally use one supervised EOS position per word.



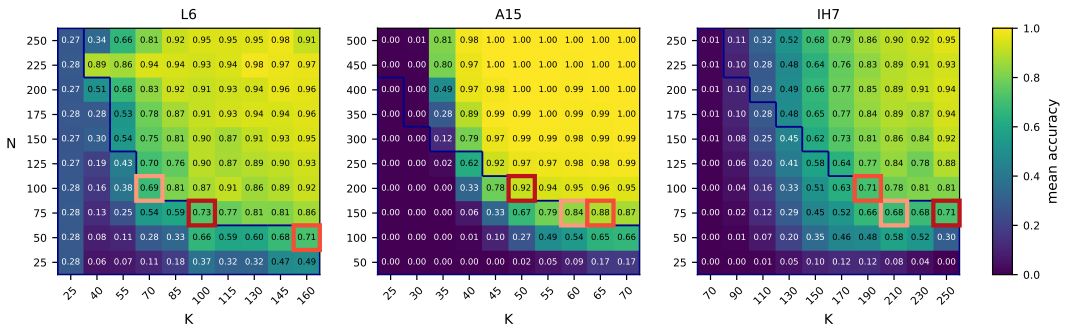
(a) Evaluation at absolute test length 500



(b) Evaluation at relative test length $5K$



(c) Here is Figure 5 for completeness. For a 0.75 minimum accuracy threshold, highlighted in red are the 3 configurations that have the smallest token-budget.



(d) For a maximum token-budget threshold, respectively of $8000, 10000, 20000$ tokens, highlighted in red are the 3 configurations that achieve the best accuracy

Figure 11: **Tradeoff between training length and data complexity**. One task of each family: L_6, A_{15} and IH_7 to study the tradeoff between the two quantities.

We also include a control experiment in which the axis labelled N specifies a budget of supervised token positions rather than a number of words. This budget includes EOS for L_k and A_k, but not

for IH_k. Gains from increasing K at a fixed budget suggest a benefit from longer contexts beyond simply supplying more supervised positions.

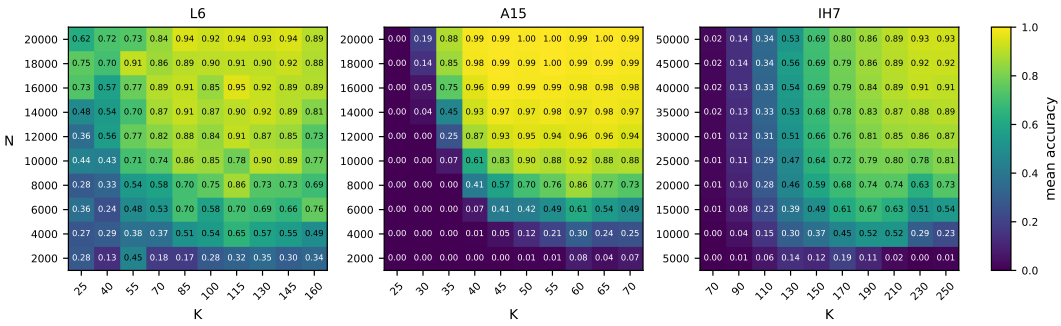
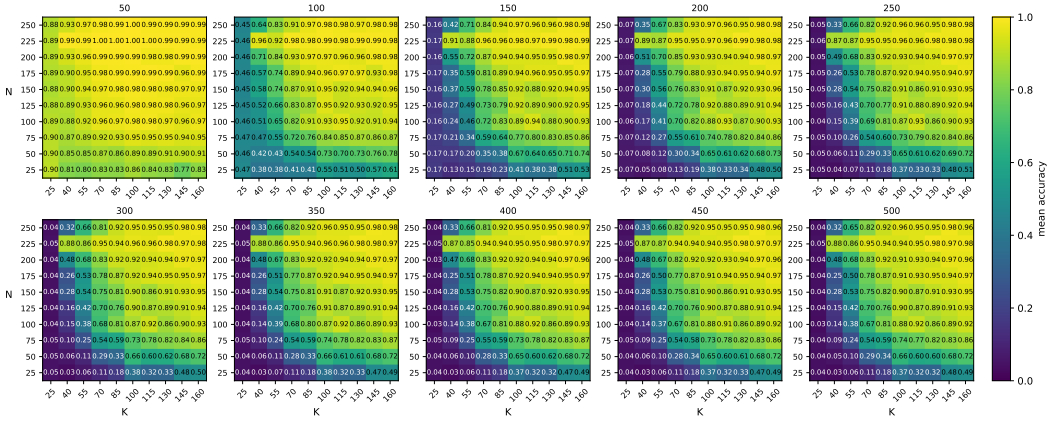
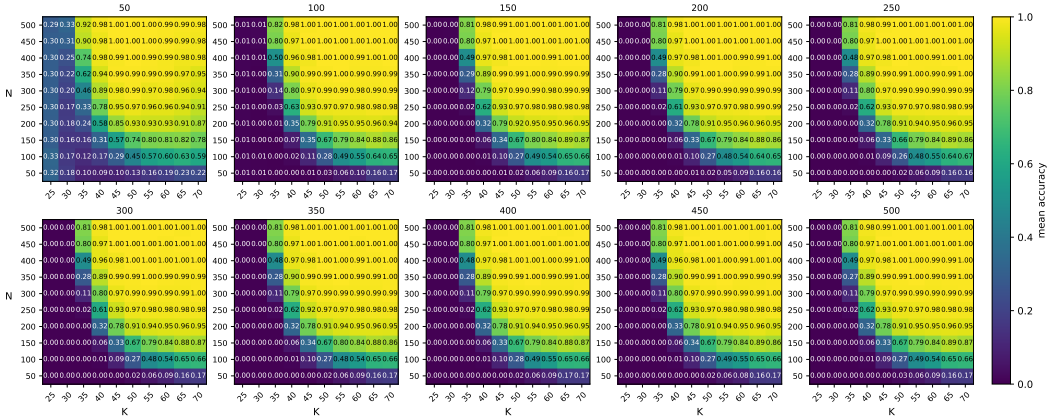


Figure 12: Control experiment for L_6, A_{15}, and IH_7. The axis labelled N gives the supervised-token budget rather than the number of words, including EOS for the first two tasks only.

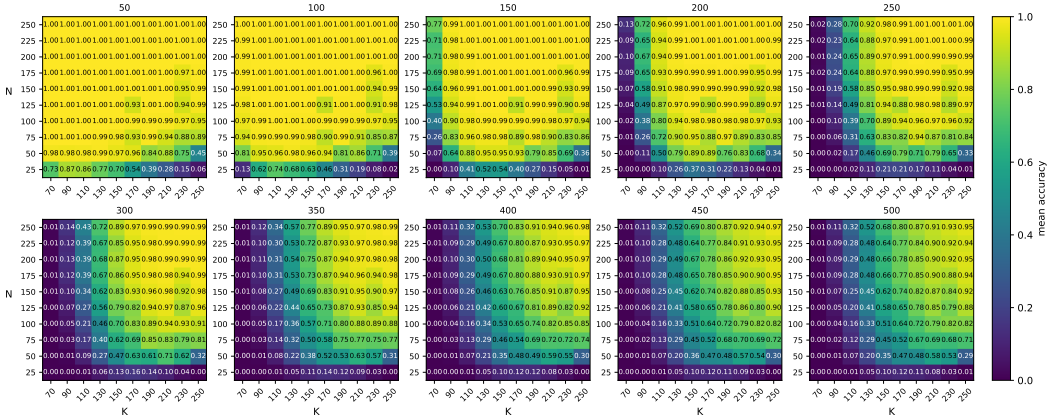
We have seen above the aggregation of the Length–Sample tradeoff of three distinct tasks L_6, A_{15}, and IH_7. We complete the collection with a compendium, of many combinations with increasing test lengths for each of three tasks.



(a) Alternating language L_6

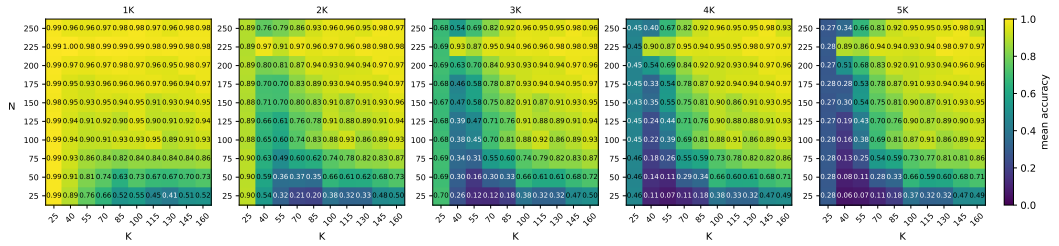


(b) Alphabet language A_{15}

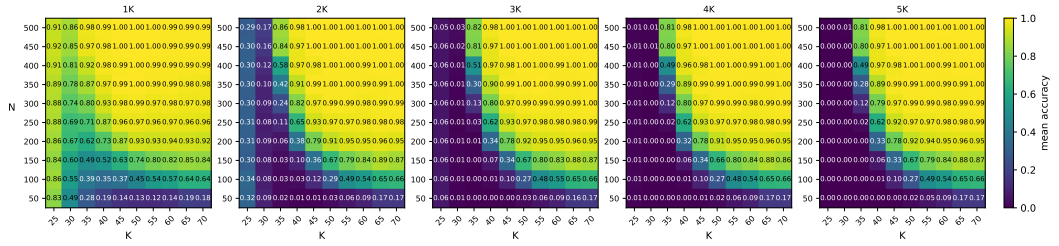


(c) Induction Head task IH_7

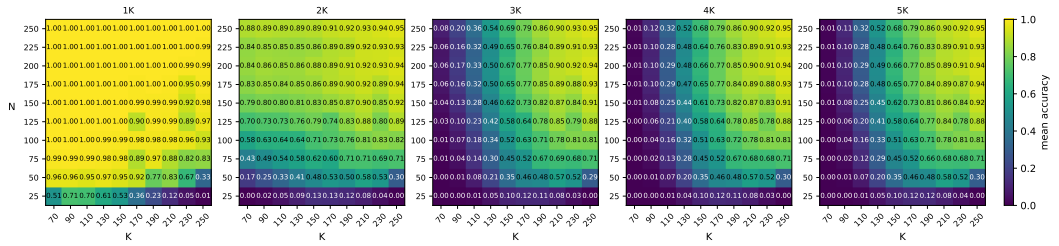
Figure 13: Tradeoff between training length and data complexity: evaluation on absolute lengths, from length 50 to length 500 with additive incrementation of 50.



(a) Alternating language L_6

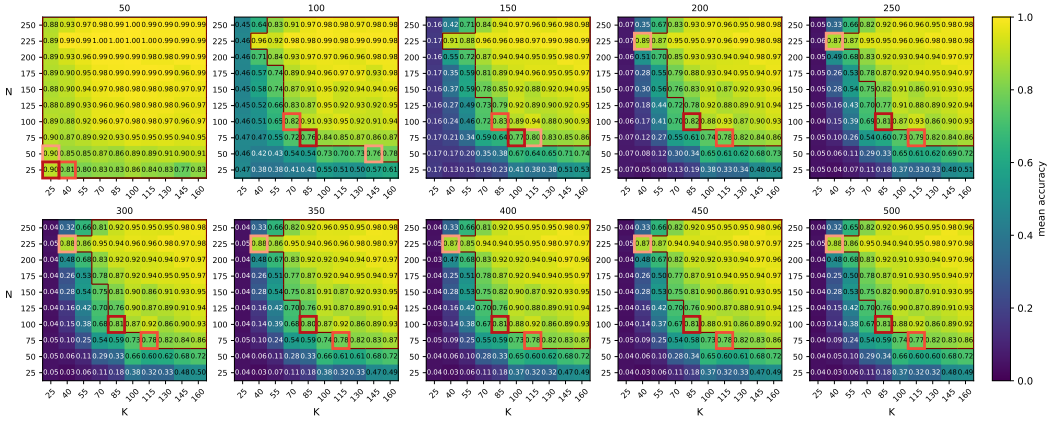


(b) Alphabet language A_{15}

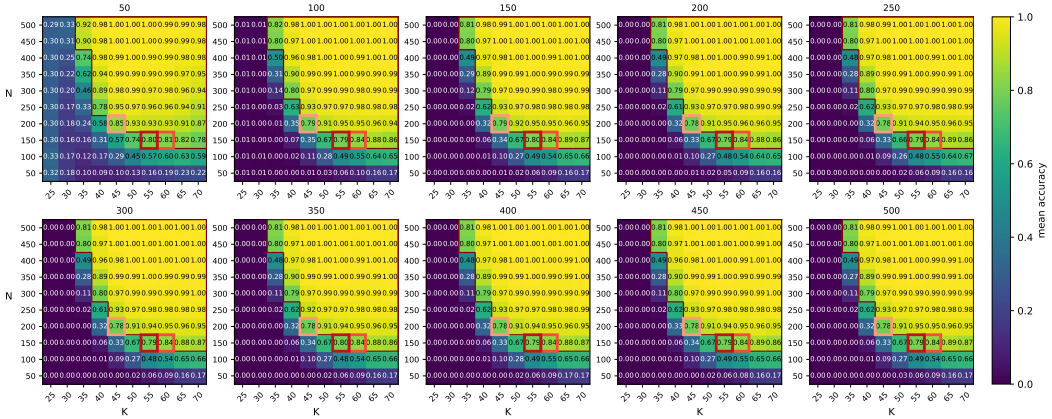


(c) Induction Head task IH_7

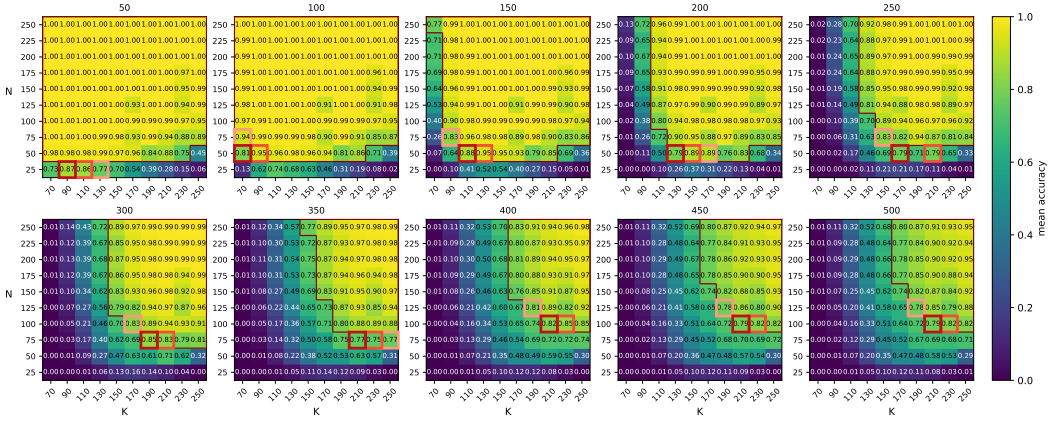
Figure 14: Tradeoff between training length and data complexity: evaluation on relative lengths $1K, 2K, 3K, 4K, 5K$, with K the training length of the involved cells configuration.



(a) Alternating language L_6

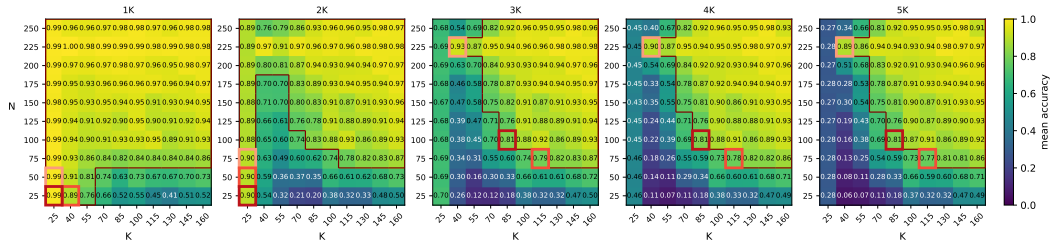


(b) Alphabet language A_{15}

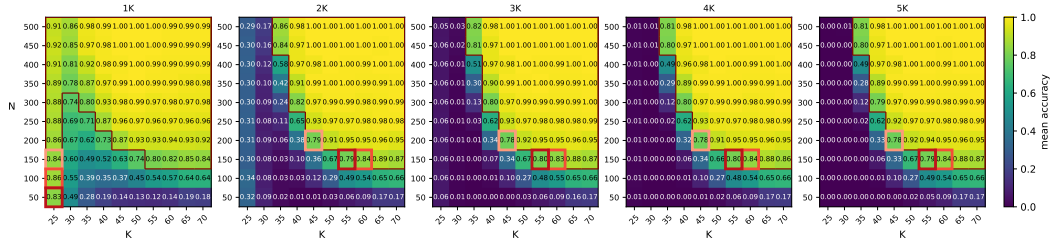


(c) Induction Head task IH_7

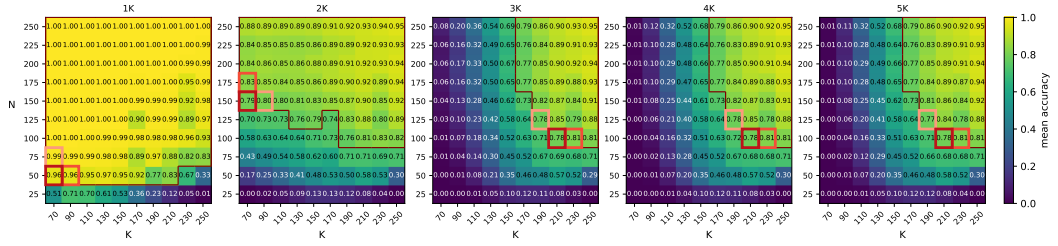
Figure 15: Tradeoff between training length and data complexity: evaluation on absolute lengths, from length 50 to length 500 with additive incrementation of 50. Red contouring for accuracy ≥ 0.75; highlighted in red are the 3 configurations under this constraint that have the smallest token-budget.



(a) Alternating language L_6

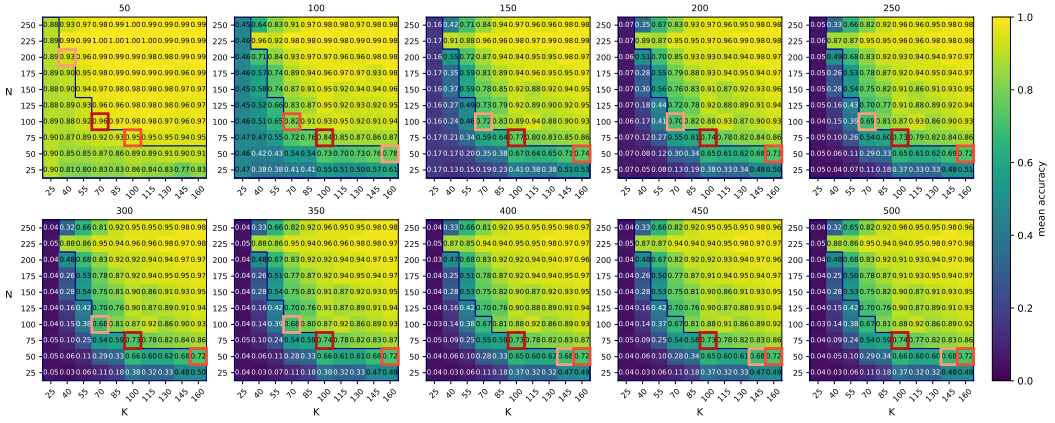


(b) Alphabet language A_{15}

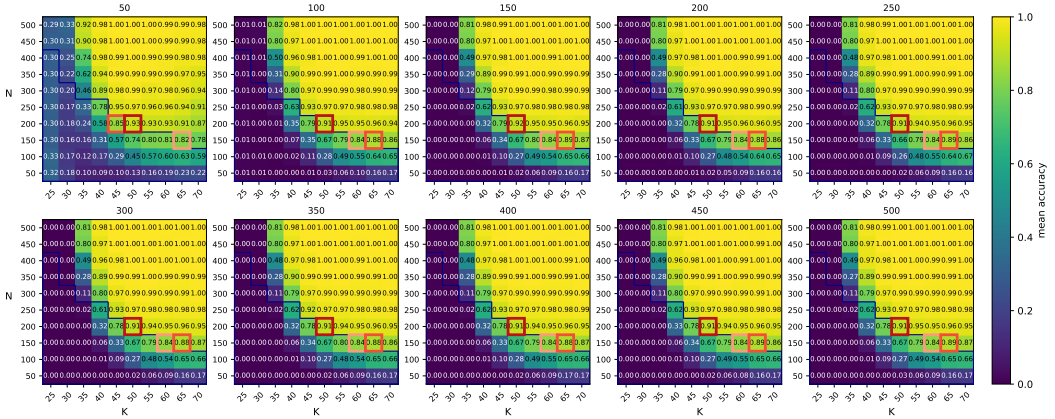


(c) Induction Head task IH_7

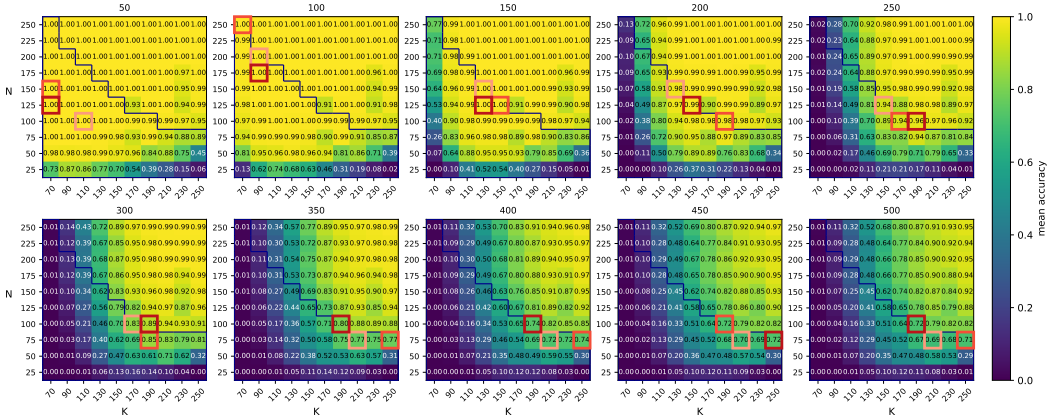
Figure 16: Tradeoff between training length and data complexity: evaluation on relative lengths $1K, 2K, 3K, 4K, 5K$, with K the training length of the involved cells configuration. Red contouring for accuracy ≥ 0.75; highlighted in red are the 3 configurations under this constraint that have the smallest token-budget.



(a) Alternating language L_6. Contouring cells with maximum token-budget 8000

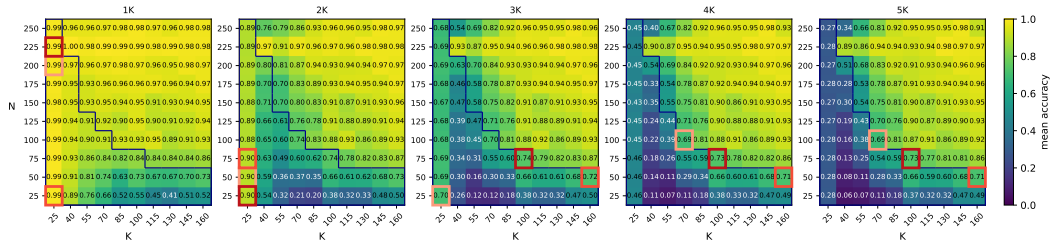


(b) Alphabet language A_{15}. Contouring cells with maximum token-budget 10000

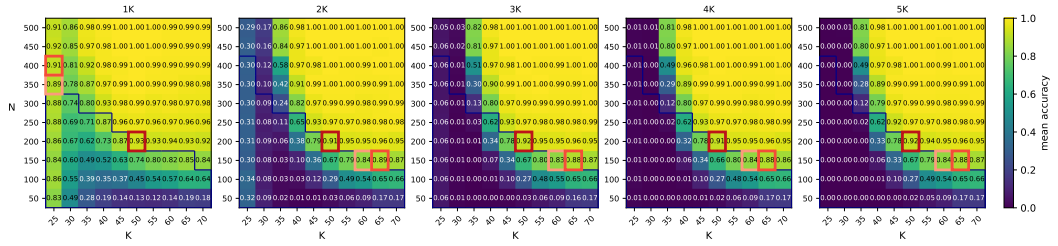


(c) Induction Head task IH_7. Contouring cells with maximum token-budget 20000

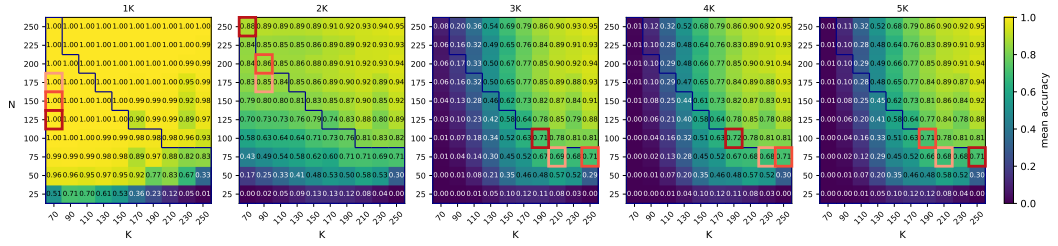
Figure 17: Tradeoff between training length and data complexity: evaluation on absolute lengths, from length 50 to length 500 with additive incrementation of 50. For their respective maximum token-budget threshold, highlighted in red are the 3 configurations that achieve the best accuracy.



(a) Alternating language L_6. Contouring cells with maximum token-budget 8000



(b) Alphabet language A_{15}. Contouring cells with maximum token-budget 10000



(c) Induction Head task IH_7. Contouring cells with maximum token-budget 20000

Figure 18: Tradeoff between training length and data complexity: evaluation on relative lengths $1K, 2K, 3K, 4K, 5K$, with K the training length of the involved cells configuration. For their respective maximum token-budget threshold, highlighted in red are the 3 configurations that achieve the best accuracy.

I AI USAGE

In this work, we used generative AI tools for: formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, assist with translation.

We have not used generative AI tools for: help develop theoretical models or conceptual frameworks, propose or refine hypotheses, design or provide feedback on research methodology or experiments, support qualitative and thematic data analysis, interpret results

and: generate synthetic data sets, implement methods, clean and reformat dataset are not applicable to this work.

Additionally, we used generative AI tools for: create or edit software code, draft parts of a research paper, sourcing/searching for information, edit a research paper to improve readability, propose a title or keywords for a research paper.

We have reviewed all AI-assisted work. For example, we checked LLM-generated research advices for potential plagiarism through a manual literature survey, we checked mathematical soundness of every mathematical claims, we tested the soundness of the LLM-generated/assisted code experiments, we did supervise and/or overwrite AI drafted writings in the paper. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.