Mechanistic Evaluation of Transformers and State-Space Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

State space models (SSMs) for language modelling promise an efficient and performant alternative to quadratic-attention Transformers, yet show variable performance on recalling basic information from the context. While performance on synthetic tasks like Associative Recall (AR) can point to this deficiency, behavioural metrics provide little information as to *why*—on a mechanistic level—certain architectures fail and others succeed. To address this, we conduct experiments on AR and find that only Transformers and Based SSM models fully succeed at AR, with Mamba a close third, whereas the other SSMs (H3, Hyena) fail. We then use causal interventions to explain why. We find that Transformers and Based learn to store key–value associations in-context using induction heads. By contrast, the SSMs compute these associations only at the last state, with only Mamba succeeding because of its short convolution component. To extend and deepen these findings, we introduce Associative Treecall (ATR), a synthetic task similar to AR based on PCFG induction. ATR introduces language-like hierarchical structure into the AR setting. We find that all architectures learn the same mechanism as they did for AR, and the same three models succeed at the task. These results reveal that architectures with similar accuracy may still have substantive differences, motivating the adoption of mechanistic evaluations.

https://anonymous.4open.science/r/tinylang-1061/

1 Introduction

Transformers with quadratic attention remain the dominant architecture in language modelling despite numerous proposed efficient alternatives. Most notably, **state-space models** (SSMs) achieve impressive perplexities and benchmark scores [e.g. Gu and Dao, 2024]. Yet, SSMs exhibit deficiencies that benchmarks often fail to capture; for example, they struggle to perform **retrieval**, i.e. copying from the context [Jelassi et al., 2024, Wen et al., 2024, Waleffe et al., 2024, Bick et al., 2025].

Controlled synthetic tasks can make these limitations clear by isolating specific capabilities and enabling expressive experimentation at small scales across architectures. Particularly, much work has used the **associative recall** (AR) task as a testbed for studying in-context retrieval across architectures. In turn, AR has informed the design of novel LM architectures [e.g. Based; Arora et al., 2024b].

Yet performance on synthetic tasks is measured solely via behavioural metrics like task accuracy. This is a missed opportunity: an advantage of these synthetic tasks is that they are designed to isolate a *specific behaviour* that implicates a mechanistic solution. For example, language models should solve AR by storing key–value associations in-context at the value, a mechanism termed the **induction**

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

head in Transformers [Olsson et al., 2022, Fu et al., 2023]. We should therefore directly check whether each architecture learns induction as part of performance evaluation on AR.

Here, we propose using tools from mechanistic interpretability to directly analyse the mechanisms used to solve synthetic tasks. We use **causal interventions** [Geiger et al., 2024] on model internals to understand how these tasks are learned and implemented across a variety of architectures (§4). This allows us to track the emergence (or lack thereof) of the correct association and retrieval mechanisms inside the model, beyond just observed task accuracy. Through comprehensive experiments on AR, we find that all SSMs except Based learn an inefficient direct-retrieval solution to AR, and that Mamba strongly relies on its short convolution component to perform AR.

To deepen our findings, we introduce **Associative Treecall** (ATR), a novel synthetic retrieval task more similar to real-world natural language retrieval than AR (§3). ATR uses a probabilistic context-free grammar (PCFG) to generate hierarchical data, on which we ask AR-like queries. Since keys and values need not be adjacent to each other, ATR requires a true non-positional retrieval mechanism, which may challenge architectures that are designed for AR. Interestingly, we observe the same mechanisms are implicated across architectures on ATR as on AR, indicating that association mechanisms are not task-dependent.

Our results offer a framework for better understanding and evaluating synthetic task performance in terms of mechanistic interpretability. Mechanistic evaluations reveal fundamental differences between architectures beyond what we learn from behavioural performance, thus serving as a new tool for architecture analysis and design.

2 Related work

Associative Recall. Associative Recall (AR)[1] is a synthetic task that evaluates in-context retrieval for language model architectures, from early work on recurrent neural networks [Graves et al., 2014, Ba et al., 2016, Danihelka et al., 2016, Zhang and Zhou, 2017] to modern SSMs [Fu et al., 2023, Poli et al., 2023, Lutati et al., 2023, Jelassi et al., 2024, Arora et al., 2024a,b, Gu and Dao, 2024, Dao and Gu, 2024, Trockman et al., 2024, Liu et al., 2024a, Okpekpe and Orvieto, 2025, Li et al., 2025b, Wang et al., 2025]. An AR task consists of a sequence of key–value pairs followed by a single *query* key; the goal is to produce the corresponding value. For example,

$$\texttt{A 2 C 3 F 9 D 1 C} \rightarrow \texttt{3} \tag{1}$$

Here, the correct next token is `3`, since it is the value associated with the key `C` in context. Despite being synthetic, AR has a direct analogue in natural language: *induction*, referring to in-context copying of sequences [Elhage et al., 2021, Olsson et al., 2022]. Arora et al. [2024a,b] show that architecture-level improvements on AR translate directly to natural-language induction.

Mechanistic interpretability. In order to measure the contribution of individual model components (neurons, layers, etc.) to output behaviour, we can apply causal interventions on neural network internals [Geiger et al., 2021, 2024]. Informally, the core idea is to overwrite an activation at a specific component using a counterfactual input. If this changes model behaviour, then that component is causally relevant to the mechanism underlying that behaviour.

Some prior work in mechanistic interpretability has studied how some language models solve in-context retrieval tasks like induction and multiple choice question answering [Olsson et al., 2022, Lieberum et al., 2023, Brinkmann et al., 2024, Wiegreffe et al., 2025, Bick et al., 2025], as well as the training dynamics of Transformers on toy tasks using mechanistic metrics [Nanda et al., 2023, Reddy, 2024, Singh et al., 2024, Edelman et al., 2024, Tigges et al., 2024, Yin and Steinhardt, 2025]. Yet thus far, *architectural comparisons* on synthetic tasks have not made use of causal interventions.

3 Synthetic retrieval tasks

Induction, wherein key–value associations are stored in-context, is the memory-efficient mechanism implicated for retrieval tasks like AR in quadratic attention Transformers. Yet AR can also be solved

[1] Also known as *associative retrieval*, *associative memory*, or *induction*.

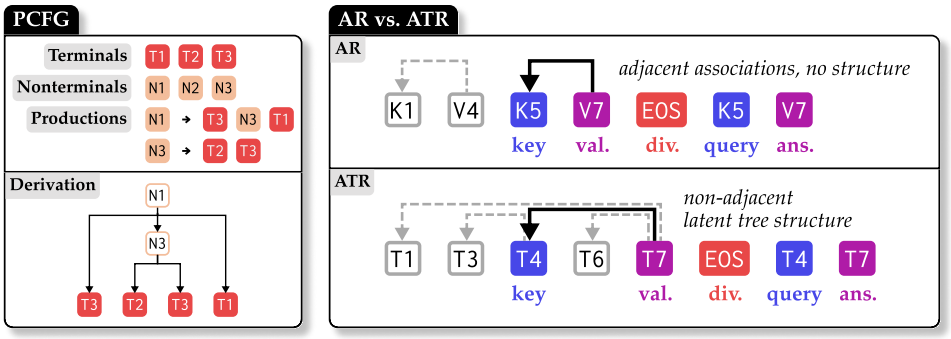


Figure 1: **PCFG**: An illustrative example of a PCFG and its components, with an example derivation (with final string) below. **AR vs. ATR**: Comparing AR and ATR using example documents; both tasks provide a document with key–value associations in-context and ask a query about one such association. However, associations in ATR need not involve adjacent tokens and are tree-structured.

through naïve positional association, and indeed SSMs theoretically learn a less efficient solution [Jelassi et al., 2024]. To elucidate this, we apply our mechanistic evaluation framework to compare architectures on *two* synthetic retrieval tasks: Associative Recall and **Associative *Tree*call** (ATR). Compared to AR, ATR is a novel language-like task with tree structure and more parameters for controlling task difficulty (§3.2). Critically, ATR cannot be solved with naïve positional association, enabling us to explicitly test if models learn different mechanisms for association in the hierarchical setting. We build upon prior work on formal-language synthetic tasks [White and Cotterell, 2021, Valvoda et al., 2022, Hahn and Goyal, 2023, Strobl et al., 2024, Allen-Zhu and Li, 2024, Akyürek et al., 2024, Pandey, 2024, Lubana et al., 2024, *inter alia*].

3.1 Associative Treecall

Since a standard AR document (eq. (1)) consists of *adjacent* key–value pairs, one can associate each key with its corresponding value solely using relative position. Yet many natural language retrieval tasks require association over latent hierarchical structure. For example:

(2) *John had chicken and Mary had pork. The chicken was eaten by* $\rightarrow$ *John*

Answering this query requires associating *John* with *chicken* and *Mary* with *pork*, and then retrieving the appropriate association for *John*. A solution employing relative positional association would not robust to the possible range of variation (*John had some chicken*, *John decided to have chicken*, etc.).

This type of retrieval is widely studied in cognitive science as *binding*. The mechanisms underlying natural-language binding in LMs have been examined by Kim and Schuster [2023], Feng and Steinhardt [2024], Prakash et al. [2024], Li et al. [2025a]. Yet no synthetic analogue of this task exists to isolate this mechanism and enable direct comparison to AR. ATR thus allows us to study how different architectures implement binding, and ask if these solutions generalize from simple AR.

An ATR corpus is drawn from a synthetic probabilistic context-free grammar (PCFG) whose parameters we set. Each document consists of a string sampled from the PCFG, with latent structure made up of **parent–child** relations between symbols, followed by a divider token (`EOS`) and a query about one such relation. The PCFG has one special property which establishes the parent–child relationships: for the right-hand side of each production rule, the rightmost symbol is always a terminal, and is the *parent* of the symbols created by this production. We sample strings by selecting an iid nonterminal and recursively applying production rules according to the PCFG distribution. We show an example in Figure 1 and formalise definitions in appendix A. Since the number of tokens separating parents and their children may vary, ATR cannot be solved by a positional associative mechanism.

3.2 Parameters

PCFG setup. For each experiment, we generate a single PCFG to use across all models to ensure fair comparisons, with parameters in Table 1. We also reject any samples that have more than 1024 symbols, which only affects the sampling distribution for the most complex PCFGs we use.

Param.	Description
H	Is the head terminal at the left or the right of each production?
d_{max}	Maximum depth permitted for the PCFG to generate.
L_{max}	Maximum number of symbols of the right-hand side of a production rule.
R_{max}	Maximum number of production rules for each nonterminal.
$\lvert\mathcal{N}\rvert$	Number of nonterminal symbols in the PCFG vocabulary.
$\lvert\Sigma\rvert$	Number of terminal symbols in the PCFG vocabulary.
r_Σ	Relative weightage on choosing a terminal when sampling production rules.

Table 1: Parameters used for constructing a PCFG. We define PCFGs in Greibach Normal Form (GNF); see Appendix A for more details.

Queries. Each PCFG sample of length n provides us with a set of $n-1$ eligible parent–child queries (i.e. a tree with $n-1$ edges). However, terminals may occur multiple times, so a query about a specific symbol may present ambiguity; thus, when presenting a query we consider it to *only* refer to the rightmost instance of that symbol.[2] Therefore, the maximum number of eligible queries over all samples is $\min(n-1, |\Sigma|)$. To minimise the ability to heuristically guess, we inversely weight parent–child pairs by the parent's child count when sampling queries.

3.3 Methodology

Datasets. We generate synthetic pretraining and evaluation datasets for both tasks. For each setting, the trainset has $100,032$ examples and the eval/dev sets have 320 examples. In AR, we use disjoint key and value vocabularies; in ATR, keys and values are both sampled from the set of terminals. In each document, we separate the document from the query with a divider token, and provide only a single query. Example AR/ATR documents are in Figure 1; further details in appendix C.

Models. We pretrain models from scratch on a variety of synthetic tasks. We use the exact architecture implementations from the `zoology`[3] library [Arora et al., 2024b], except for behaviour-preserving modification of the LM backbone to enable interventions with `pyvene`[4] [Wu et al., 2024] on the sequence mixers, MLPs, and layer blocks. The LM backbone for all architectures is the same, with pre-norm blocks of alternating sequence mixers and MLPs (except for Mamba, which has no MLP) followed by LayerNorm at the end. We experiment with the following architectures: Attention [Vaswani et al., 2017], BaseConv [Arora et al., 2024a], Based [Arora et al., 2024b], H3 [Fu et al., 2023], Hyena [Poli et al., 2023], and Mamba [Gu and Dao, 2024]; further details on model configurations are given in appendix B.

Training. We minimise cross-entropy loss, and mask the loss on all tokens except the query (the underlined token in the example below). We use the AdamW optimiser with $\beta = (0.9, 0.999), \epsilon = 10^{-8}$ and no weight decay. We warm up learning rate for the first 10% of training and then follow a cosine decay schedule to 0 for the remainder of training. We train for either 16 epochs (on AR) or 32 epochs (on ATR) with a batch size of 32. Each experiment trains ≈ 200 models over all hyperparameters. Runtime varies from 0.5 to 5 hours, depending on hardware, task, and architecture. Overall, we used $< 10,000$ GPU-hours in total, on a cluster with various NVIDIA machines (with GPU memory ranging from 12.3G to 143.8G).

Behavioural metrics. We report behavioural metrics given the model's predicted probabilities over the vocabulary $\hat{\mathbf{y}} \in \mathbb{R}^{|\Sigma|}$ and the index of the single true answer i. Our main metric is accuracy: $\mathbb{1}[\arg\max(\hat{\mathbf{y}}) = i]$. Additionally, we compute but do not primarily report likelihood $\hat{\mathbf{y}}_i$.

4 Mechanistic metrics for AR and ATR

Behavioural metrics provide little information as to *why* certain architectures succeed or fail on tasks of interest. Mechanistic metrics, which directly measure how information flows across model compo-

[2] This is the same setup as AR with rewrites [Rodkin et al., 2025].

[3] https://github.com/HazyResearch/zoology

[4] https://github.com/stanfordnlp/pyvene

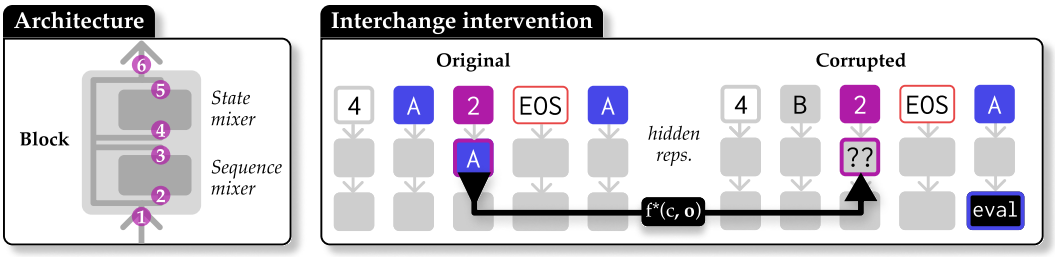


Figure 2: Our interchange intervention setup for analysing AR and ATR. **Left**: We intervene on input and output representations of whole blocks (1 and 6), sequence mixers (e.g. attention blocks; 2 and 3), and state mixers (4 and 5). **Right**: An example intervention on AR where we corrupt and attempt to restore the *key* (A) by intervening at the *value* token in an intermediate representation. We evaluate the downstream effect on the next-token prediction at the *query*.

nents and token positions, can tell us how AR and ATR are being solved by different architectures, and thus help us understand failures. We illustrate our approach in Figure 2.

We use interchange interventions [Geiger et al., 2021, 2024] to understand and measure how solutions to AR and ATR are implemented across architectures. We introduce this operation and define the resulting metrics for our tasks below. Our implementation uses the pyvene library [Wu et al., 2024].

Interchange intervention. Consider a language model $p(\cdot)$ and some input $\mathbf{b}$. We select a component f inside that model which computes some internal representation $f(\mathbf{b})$ during the LM's forward pass. Now, consider a counterfactual input $\mathbf{s}$: this produces a counterfactual representation $f(\mathbf{s})$ when processed by f. We want to understand what about the output of p is dependent on f. Therefore, we perform an intervention which replaces the output $f(\mathbf{b})$ with that of $f(\mathbf{s})$ during the computation of $p(\mathbf{b})$, with the change propagating downstream. The result is notated $p_{f \leftarrow f^*}(\mathbf{b}, \mathbf{s})$.

Concrete setup for AR and ATR. We take $\mathbf{o}$ to be a ground-truth document from our data distribution and $\mathbf{c}$ to be a version of that document with exactly one important token corrupted: the *key* (see Figure 2). This corruption significantly reduces task accuracy for both AR and ATR by removing information that is necessary to answer the query.

We intervene at both the input and output each of the following model components f: each layer block, each sequence-mixer, and each state-mixer (i.e. MLP, except in Mamba which lacks this component); see Figure 2, left. We measure to what extent the intervention can restore the likelihood of the correct answer to the query, i.e. we compare restored likelihood $p_{f \leftarrow f^*}(y_{\text{true}} \mid \mathbf{c}, \mathbf{o})$ with original likelihood $p(y_{\text{true}} \mid \mathbf{o})$ and corrupted likelihood $p(y_{\text{true}} \mid \mathbf{c})$.

Metrics. Given the above three quantities, we compute **attribution score**, or what proportion of the original likelihood was restored by the intervention:

$$\text{Attrib}(f) = \frac{p_{f \leftarrow f^*}(y_{\text{true}} \mid \mathbf{b}, \mathbf{s}) - p(y_{\text{true}} \mid \mathbf{b})}{p(y_{\text{true}} \mid \mathbf{s}) - p(y_{\text{true}} \mid \mathbf{b})} \tag{3}$$

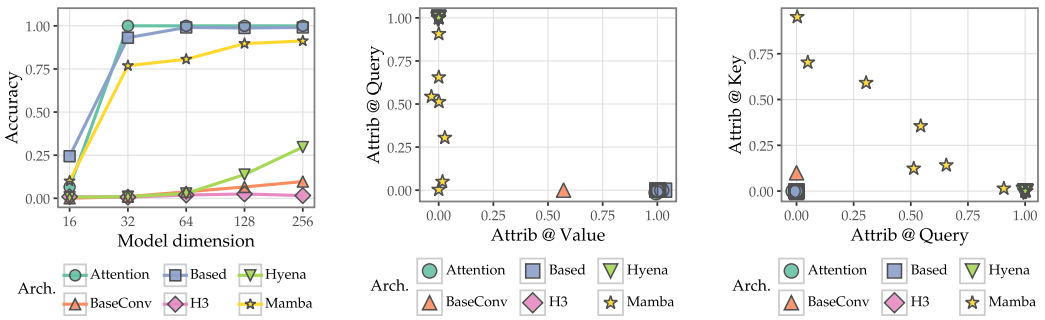
For AR and ATR in particular, there are two choices for f which help us distinguish the mechanism underlying task success. To check whether induction is the underlying mechanism, we compute metrics for f being the layer 1 *block input* at the *value* token. Alternatively, we check whether other tokens at layer 1 block input mediate information flow, indicating some sort of association-less direct retrieval mechanism: the *key*, *query*, and *divider*.

5 Experiments

We now deploy our mechanistic metrics (§4) on both AR and ATR (§3). We follow the methodology outlined in §3.3 to create a variety of AR and ATR datasets and train models with various architectures.

5.1 (Most) SSMs do not learn induction to solve AR

We run experiments on a relatively simple AR task and show that interchange interventions empirically confirm the same mechanisms underlying AR as proposed in existing theoretical work. We fix



(a) Model dimensionality vs. accuracy after tuning learning rate for each setting.

(b) Attribution score when restoring the key at its *value* (induction) vs. at the *query*, for all LRs.

(c) Same as left but with restoring at the *query* (layer 0 direct retrieval) vs. *key* (layer 1).

Figure 3: **Associative recall**: Accuracy and interchange intervention results on AR with vocabulary size 8192 and key–value count of 32. SSMs (except for Based) and Transformers learn different mechanisms.

the total number of unique keys and values in the vocabulary to be 8192, and present 32 key–value pairs in context. Our trainset includes 100032 examples. We vary model dimensionality in $\{16, 32, 64, 128, 256\}$ and sweep LR in the range $[3 \cdot 10^{-5}, 3 \cdot 10^{-2}]$ for each architecture.

Behavioural results. Figure 3a demonstrates that task accuracy on AR cleanly separates Attention, which achieves 100% accuracy at $d \geq 32$, from nearly all SSMs. Based solves AR near-perfectly with roughly the same dimension-wise scaling curve as Attention, achieving a maximum accuracy of 99.06%. However, Mamba is a close third and clearly better than other SSMs at AR, albeit achieving a less-than-perfect 91.25% at $d = 256$.

Mechanistic analysis. We compute Attrib for layer 1 block input at the *value* token vs. *query* token for all training runs where $p(y_{\text{true}} \mid \mathbf{o}) - p(y_{\text{true}} \mid \mathbf{c}) > 0.01$.[5] A high attribution score on the *value* token indicates **induction** as the underlying mechanism while *query* indicates **direct retrieval** at the final state, performed in layer 0. Our results in Figure 3b cleanly separate Attention (with nearly all checkpoints with 100.00% attribution at the *value*) and Based, which only perform induction, from other SSMs, which perform direct retrieval. While only a single BaseConv checkpoint passes our filter, it has the greatest attribution score on the *value*, indicating an induction mechanism.

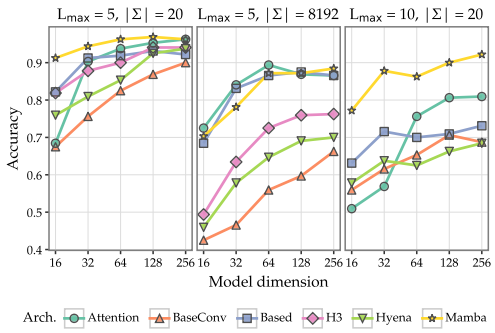
SSMs perform direct retrieval at varying layers: the best-performing Mamba, Hyena, and H3 models almost entirely perform direct retrieval at layer 0 via the *query* token, while worse SSM checkpoints use a mix of *query* and *key* tokens, indicating delayed direct retrieval by both layer 0 and layer 1. Jelassi et al. [2024] shows that direct retrieval in SSMs has asymptotically worse capacity than the induction solution, and this is reflected in performance on AR.

5.2 Per-architecture mechanisms are similar between ATR and AR

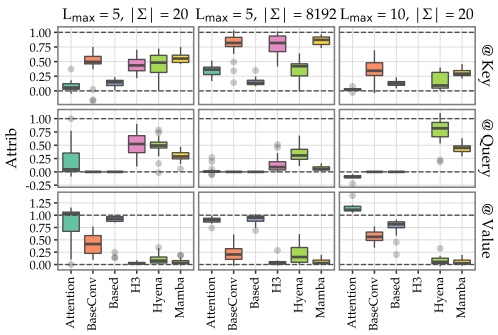
We consider four initial settings to study models on ATR, over all combinations of $L_{\text{max}} = \{5, 10\}$ and $|\Sigma| = \{20, 8192\}$. We keep all other parameters fixed with settings given in appendix C. Varying L_{max} controls the possible distances between keys and values in the PCFG sample without affecting other properties that play a role in task difficulty (e.g. depth). Varying $|\Sigma|$ stresses the state capacity, since more key–value pairs must be tracked, without affecting syntactic complexity. We sweep the same model dimensionalities as in §5.1, and a smaller learning rate range of $[3 \cdot 10^{-5}, 3 \cdot 10^{-3}]$.

Behavioural results. We report results in Figure 4a. Surprisingly, Mamba is highly successful at ATR. On the small terminal count setting ($|\Sigma| = 20$) Mamba matches or outperforms all other architectures at all model dimensions, particularly with longer production rules ($L_{\text{max}} = 10$) with performance of 92.19% vs. 80.94% for Attention at $d = 256$. This is particularly surprising because longer production rules imply greater positional variation between keys and values, which ought to

[5] We filter in order to discard low-performing and noisy runs.



(a) Model dimensionality vs. accuracy after tuning learning rate for each setting.



(b) Summarised attribution scores at *key*, *value*, and *query* for each setting, when restoring the key.

Figure 4: **Associative Treecall**: Accuracy and interchange intervention results on ATR across varying settings. The same trend as on AR holds, with Attention, Based, and Mamba achieving high performance but with entirely different mechanisms.

stress AR-focused SSM designs. Attention only manages to outperform Mamba slightly on the large terminal count setting ($|\Sigma| = 8192$) when $d \leq 64$.

Mechanistic analysis. We conduct the same analysis as for AR. We recover the same overall trends but with greater inter-architecture variance: Figure 4b shows that Attention, Based, and BaseConv all primarily learn induction mechanisms, whereas the remaining SSMs perform direct retrieval as on AR, with high attribution scores on either the *key* (indicating direct retrieval by the layer 1 sequence mixer) or the *query* (indicating the same but by layer 0).

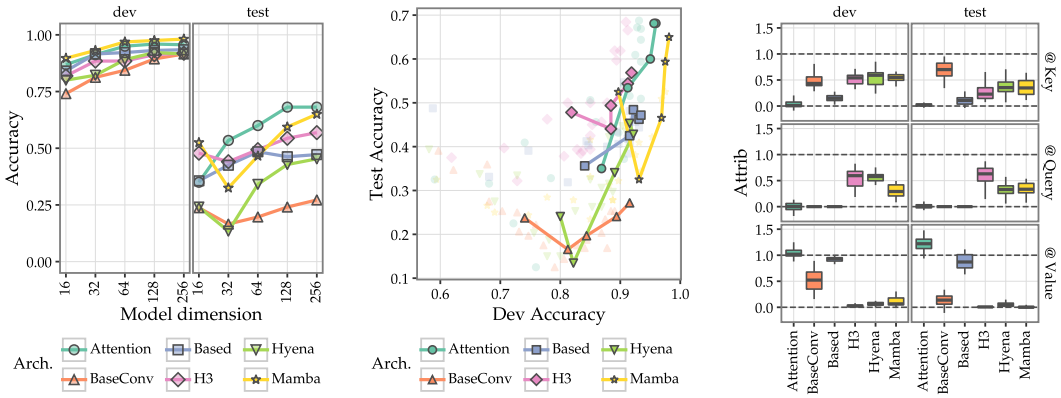
Intriguingly, Figure 4b shows that different SSMs form different strategies across task difficulties; in particular, all direct-retrieval SSMs favour delaying retrieval to layer 1 when terminal count is large ($|\Sigma| = 8192$), but use a mix of layers otherwise. Regardless, the same tendency from AR recurs: SSMs besides Based and BaseConv do not perform induction, but Mamba is still highly performant. Strikingly, as the next section shows, Mamba also achieves high generalization performance on ATR.

5.3 Mamba's solution to ATR does generalise

We reuse the easiest settings from our ATR experiment ($L = 5, |\Sigma| = 20$) and construct a new dataset with a train–test split on query–answer pairs. Specifically, 80% of possible unique query–answer pairs are provided in the training set, while 20% are only in the test set and thus never trained on. We seek to assess whether models learn a general mechanism for parent–child relations in ATR or if the impressive results of Mamba (as well as Attention and Based) are merely the result of better memorisation of the PCFG parameters. This setup is akin to Wang et al. [2024]'s technique of train–test split on multi-hop queries; we provide supervision on individual query and answer types, but not on some compositions of them.

Behavioural results. We select the checkpoint with the highest dev accuracy for each architectural and dimensionality setting, after sweeping LR. We plot the dev and test accuracies of each of these checkpoints in Figure 5a; all models have much lower test accuracy (e.g. Attention with $d = 256$ has 95.62% dev and 68.12% test accuracy). Attention achieves the greatest dev accuracies on $d \geq 32$. Mamba's relative ranking is lower than on the in-distribution setting in §5.2, but it still achieves the overall second-highest dev accuracy (65.00% at $d = 128$). Surprisingly, H3 generalises well despite its poor dev accuracy, beating Mamba on test accuracy in 3 out of 5 settings.

We compare dev and test accuracies across all LRs in Figure 5b. We find that while Mamba does have unusually high dev accuracy given a selected test accuracy (indicating greater memorisation than models with other architectures), its dev accuracy is still generally higher than non-Attention architectures. Interestingly, H3 has nearly Attention-level generalisation while BaseConv exhibits vanishingly little generalisation. Overall, behavioural metrics show that Mamba does nontrivially generalise on ATR, albeit not as well as Attention.



(a) Model dimensionality vs. accuracy on checkpoints with highest dev accuracy.

(b) Dev vs. test accuracy, with highest dev accuracy checkpoints at each dim. highlighted.

(c) Attribution scores for all checkpoints (except outliers), compared between dev and test sets.

Figure 5: **Generalisation on Associative Treecall**: Accuracy and interchange intervention results on ATR with train–test split. Scores are reported on dev (with in-distribution query–answer pairs from training) and test (OOD). We highlight the checkpoint with the best dev score in each setting.

Mechanistic analysis. We report a summary of attribution scores at different tokens (*key*, *query*, *value*), comparing on dev and test sets across all checkpoints in Figure 5c. We find largely consistent mechanisms underlying behaviour on both dev and test, and these match attribution scores on ATR without train–test split. The only exception is that BasedConv does induction on the dev set but not nearly as much on the test set; its induction mechanism is more brittle than Attention and Based.

Overall, the induction mechanism is not more general than the direct retrieval mechanism; both Attention and Mamba show greater generalisation than other architectures despite their entirely different solutions, and our mechanistic evaluations confirm that this solution is consistent across in-distribution and out-of-distribution queries.

5.4 Short convolutions enable AR and ATR in Mamba and Based

Throughout all our experiments on AR and ATR, we repeatedly observed that Attention, Based, and Mamba are the highest-performing architectures. However, their underlying mechanisms differ: Attention and Based learn **induction**, a 2-layer mechanism which stores key–value associations at the value token as an intermediate step, whereas Mamba uses **direct retrieval**, a 1-layer mechanism which directly writes an association to the query token.

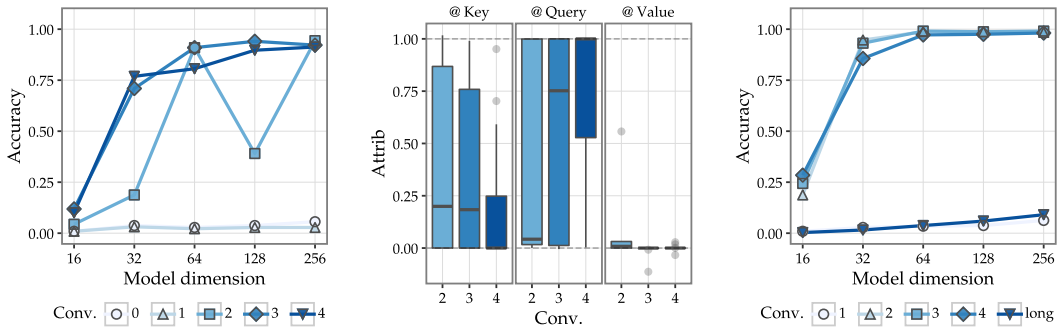
Importantly, Based and Mamba share a key architectural component: **short convolutions**. We hypothesise that this component is necessary[6] for performing association (as in AR and ATR) when using a subquadratic sequence mixer. We conduct experiments on AR where we shorten the convolution kernel size in Mamba (from the default $d_{\text{conv}} = 4$ to $\{3, 2, 1\}$, and deleting it) and replace the Based short convolution with implicitly-parametrised long convolution [Poli et al., 2023].

Results. We report results of our ablations in Figure 6. On Mamba (Figure 6a), we find a step change in task accuracy when increasing d_{conv} from 1 to 2, which introduces previous token information and thus enables AR. Without short convolution, Mamba fails to learn AR. Figure 6b further shows that larger kernel size leads to earlier (in layer 0) direct retrieval. Finally, besides $d_{\text{conv}} < 2$ like Mamba, implicit long convolution in Based also significantly harms AR performance (Figure 6c). Therefore, we conclude that short convolutions are responsible for association on AR in Mamba and Based.

6 Discussion

Why mechanistic evaluations over behavioural metrics? Architectural advances on language modelling are largely uncovered and presented in an empirical manner; beyond intuition, we have

[6] Since Hyena also has a short convolution, this may not be *sufficient* for good performance on association.



(a) Accuracy on AR (length 32) for **Mamba** when varying the kernel size of the short convolution.

(b) Summarised attribution scores across all checkpoints for **Mamba** when varying conv. kernel size.

(c) Accuracy on AR for **Based** when varying conv. kernel size or using implicit long conv.

Figure 6: **Ablating short convolution**: Accuracy and interchange intervention results when ablating parameters of the short convolution component in Mamba, Based, and BaseConv.

little justification as to *why* a modification or innovation improves model performance. Synthetic tasks already regularly inform progress on subquadratic architecture design (such as SSMs), but treating such tasks as another downstream evaluation is loses useful signal; control over task parameters presents an opportunity to explain performance using interpretability.

ATR indicates induction is highly general. We introduced ATR to break the naïve key–value adjacency of AR, and see whether general mechanisms underlying association still emerge across architectures. We find the same induction mechanism, where the association is computed and stored at the value before retrieval, in Attention and Based for both tasks. While Olsson et al. [2022] and later works define induction on adjacent tokens, ATR is evidence that a *position-independent* and generalising (§5.3) notion of association can be implemented by a single attention head. Further investigation of ATR (e.g. multi-hop queries) is necessary to understand the limits of induction.

Short convolutions are key to association in SSMs. We showed that Mamba and Based rely on short convolutions to learn how to associate keys and values on AR and ATR. Several earlier works point to the importance of short convolution: Arora et al. [2024b] empirically show its utility on AR (along with sliding-window attention), Allen-Zhu and Alfarano [2025] introduce a short convolution component (Canon) in various architectures to improve synthetic and real task performance, and Olsson et al. [2022] show that 1-layer attention can learn induction if augmented with a length-2 convolution; further see Liu et al. [2024b], Dolga et al. [2024], Fu et al. [2023], Poli et al. [2023].

7 Limitations

While we proposed mechanistic evaluations as a new tool, behavioural metrics like accuracy are still needed to properly contextualise results. Additionally, here we did not perform mechanistic evaluation of subcomponents of sequence mixers (e.g. the selective SSM component within Mamba), due to implementation difficulties when applying interventions within hardware-optimised operators, which are inaccessible via PyTorch hooks. Finally, we focus on synthetic tasks throughout this work; extending our analyses to real-world models would help paint a more complete picture of the differences in capabilities (and underlying mechanisms) of different architectures on real-world tasks.

8 Conclusion

In this work, we introduce mechanistic evaluations as a powerful framework for comparing model architectures. This approach goes beyond high-level behavioural metrics, revealing substantive differences between architectures. Through analysis of synthetic in-context retrieval tasks, we uncover the underlying mechanisms that explain the success and failure points of various architectures. Mechanistic evaluations thus provide a useful tool for architecture design and analysis, as well as a new opportunity for interpretability research to open the blackbox of progress in AI.

References

Ekin Akyürek, Bailin Wang, Yoon Kim, and Jacob Andreas. In-context language learning: Architectures and algorithms. In *Forty-first International Conference on Machine Learning, ICML 2024*, Vienna, Austria, 2024. OpenReview.net. URL https://openreview.net/forum?id=3Z9CRr5srL.

Zeyuan Allen-Zhu and Alberto Alfarano. Physics of Language Models: Part 4.1, Architecture design and the magic of Canon layers. *SSRN*, 2025. URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5240330.

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of Language Models: Part 1, Learning hierarchical language structures. *arXiv:2305.13673*, 2024. URL https://arxiv.org/abs/2305.13673.

Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Zou, Atri Rudra, and Christopher Ré. Zoology: Measuring and improving recall in efficient language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024a. URL https://openreview.net/forum?id=LY3ukUANko.

Simran Arora, Sabri Eyuboglu, Michael Zhang, Aman Timalsina, Silas Alberti, James Zou, Atri Rudra, and Christopher Ré. Simple linear attention language models balance the recall-throughput tradeoff. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024b. URL https://openreview.net/forum?id=e93ffDcpH3.

Jimmy Ba, Geoffrey E. Hinton, Volodymyr Mnih, Joel Z. Leibo, and Catalin Ionescu. Using fast weights to attend to the recent past. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4331–4339, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/9f44e956e3a2b7b5598c625fcc802c36-Abstract.html.

Aviv Bick, Eric Xing, and Albert Gu. Understanding the skill gap in recurrent language models: The role of the gather-and-aggregate mechanism. *arXiv:2504.18574*, 2025. URL https://arxiv.org/abs/2504.18574.

Jannik Brinkmann, Abhay Sheshadri, Victor Levoso, Paul Swoboda, and Christian Bartelt. A mechanistic analysis of a transformer trained on a symbolic multi-step reasoning task. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4082–4102, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.242. URL https://aclanthology.org/2024.findings-acl.242/.

Ivo Danihelka, Greg Wayne, Benigno Uria, Nal Kalchbrenner, and Alex Graves. Associative long short-term memory. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1986–1994. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/danihelka16.html.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. *arXiv:2405.21060*, 2024. URL https://arxiv.org/abs/2405.21060.

Rares Dolga, Lucas Maystre, Marius Cobzarenco, and David Barber. Latte: Latent attention for linear time transformers. *arXiv:2402.17512*, 2024. URL https://arxiv.org/abs/2402.17512.

Ezra Edelman, Nikolaos Tsilivis, Benjamin L. Edelman, Eran Malach, and Surbhi Goel. The evolution of statistical induction heads: In-context learning markov chains. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024*, Vancouver, BC, Canada, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/75b0edb869e2cd509d64d0e8ff446bc1-Abstract-Conference.html.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. URL https://transformer-circuits.pub/2021/framework/index.html.

Jiahai Feng and Jacob Steinhardt. How do language models bind entities in context? In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=zb3b6oKO77.

Daniel Y. Fu, Tri Dao, Khaled Kamal Saab, Armin W. Thomas, Atri Rudra, and Christopher Ré. Hungry hungry hippos: Towards language modeling with state space models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=COZDy0WYGg.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 9574–9586. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/4f5c422f4d49a5a807eda27434231040-Paper.pdf.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, and Thomas Icard. Causal abstraction: A theoretical foundation for mechanistic interpretability. *arXiv:2301.04709*, 2024. URL https://arxiv.org/abs/2301.04709.

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv:1410.5401*, 2014. URL https://arxiv.org/abs/1410.5401.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv:2312.00752*, 2024. URL https://arxiv.org/abs/2312.00752.

Michael Hahn and Navin Goyal. A theory of emergent in-context learning as implicit structure induction. *arXiv:2303.07971*, 2023. URL https://arxiv.org/abs/2303.07971.

Samy Jelassi, David Brandfonbrener, Sham M. Kakade, and Eran Malach. Repeat after me: Transformers are better than state space models at copying. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=duRRoGeoQT.

Najoung Kim and Sebastian Schuster. Entity tracking in language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3835–3855, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.213. URL https://aclanthology.org/2023.acl-long.213/.

Belinda Z. Li, Zifan Carl Guo, and Jacob Andreas. (How) do language models track state? *arXiv:2503.02854*, 2025a. URL https://arxiv.org/abs/2503.02854.

Mingchen Li, Xuechen Zhang, Yixiao Huang, and Samet Oymak. On the power of convolution-augmented transformer. In Toby Walsh, Julie Shah, and Zico Kolter, editors, *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 18393–18402. AAAI Press, 2025b. doi: 10.1609/AAAI.V39I17.34024. URL https://doi.org/10.1609/aaai.v39i17.34024.

Tom Lieberum, Matthew Rahtz, János Kramár, Neel Nanda, Geoffrey Irving, Rohin Shah, and Vladimir Mikulik. Does circuit analysis interpretability scale? evidence from multiple choice capabilities in chinchilla. *arXiv preprint arXiv:2307.09458*, 2023.

Bo Liu, Rui Wang, Lemeng Wu, Yihao Feng, Peter Stone, and Qiang Liu. Longhorn: State space models are amortized online learners. *arXiv:2407.14207*, 2024a. URL https://arxiv.org/abs/2407.14207.

Zicheng Liu, Siyuan Li, Li Wang, Zedong Wang, Yunfan Liu, and Stan Z. Li. Short-long convolutions help hardware-efficient linear attention to focus on long sequences. In *Forty-first International Conference on Machine Learning, ICML 2024*, Vienna, Austria, 2024b. OpenReview.net. URL https://openreview.net/forum?id=TRrXkVdhwi.

Ekdeep Singh Lubana, Kyogo Kawaguchi, Robert P. Dick, and Hidenori Tanaka. A percolation model of emergence: Analyzing transformers trained on a formal language. *arXiv:2408.12578*, 2024. URL https://arxiv.org/abs/2408.12578.

Shahar Lutati, Itamar Zimerman, and Lior Wolf. Focus your attention (with adaptive IIR filters). In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12538–12549, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.772. URL https://aclanthology.org/2023.emnlp-main.772/.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations, ICLR 2023*, Kigali, Rwanda, 2023. OpenReview.net. URL https://openreview.net/forum?id=9XFSbDPmdW.

Franz Nowak and Ryan Cotterell. A fast algorithm for computing prefix probabilities. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 57–69, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-short.6. URL https://aclanthology.org/2023.acl-short.6/.

Destiny Okpekpe and Antonio Orvieto. Revisiting associative recall in modern recurrent models. In *First Workshop on Scalable Optimization for Efficient and Adaptive Foundation Models*, 2025. URL https://openreview.net/pdf?id=CcqAd5RPk5.

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

Rohan Pandey. gzip predicts data-dependent scaling laws. *arXiv:2405.16684*, 2024. URL https://arxiv.org/abs/2405.16684.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y. Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 28043–28078. PMLR, 2023. URL https://proceedings.mlr.press/v202/poli23a.html.

Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. Fine-tuning enhances existing mechanisms: A case study on entity tracking. In *The Twelfth International Conference on Learning Representations, ICLR 2024*, Vienna, Austria, 2024. OpenReview.net. URL https://openreview.net/forum?id=8sKcAWOf2D.

Gautam Reddy. The mechanistic basis of data dependence and abrupt learning in an in-context classification task. In *The Twelfth International Conference on Learning Representations, ICLR 2024*, Vienna, Austria, 2024. OpenReview.net. URL https://openreview.net/forum?id=aN4Jf6Cx69.

Ivan Rodkin, Yuri Kuratov, Aydar Bulatov, and Mikhail Burtsev. Associative recurrent memory transformer. *arXiv:2407.04841*, 2025. URL https://arxiv.org/abs/2407.04841.

Aaditya K. Singh, Ted Moskovitz, Felix Hill, Stephanie C. Y. Chan, and Andrew M. Saxe. What needs to go right for an induction head? A mechanistic study of in-context learning circuits and their formation. In *Forty-first International Conference on Machine Learning, ICML 2024*, Vienna, Austria, 2024. OpenReview.net. URL https://openreview.net/forum?id=O8rrXl71D5.

Lena Strobl, William Merrill, Gail Weiss, David Chiang, and Dana Angluin. What formal languages can transformers express? A survey. *Transactions of the Association for Computational Linguistics*, 12:543–561, 2024. doi: 10.1162/tacl_a_00663. URL https://aclanthology.org/2024.tacl-1.30/.

Curt Tigges, Michael Hanna, Qinan Yu, and Stella Biderman. LLM circuit analyses are consistent across training and scale. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024*, Vancouver, BC, Canada, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/47c7edadfee365b394b2a3bd416048da-Abstract-Conference.html.

Asher Trockman, Hrayr Harutyunyan, J. Zico Kolter, Sanjiv Kumar, and Srinadh Bhojanapalli. Mimetic initialization helps state space models learn to recall. *arXiv:2410.11135*, 2024. URL https://arxiv.org/abs/2410.11135.

Josef Valvoda, Naomi Saphra, Jonathan Rawski, Adina Williams, and Ryan Cotterell. Benchmarking compositionality with formal languages. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na, editors, *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6007–6018, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics. URL https://aclanthology.org/2022.coling-1.525/.

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

Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norick, Vijay Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, Sudhakar Singh, Deepak Narayanan, Garvit Kulshreshtha, Vartika Singh, Jared Casper, Jan Kautz, Mohammad Shoeybi, and Bryan Catanzaro. An empirical study of Mamba-based language models. *arXiv:2406.07887*, 2024. URL https://arxiv.org/abs/2406.07887.

Boshi Wang, Xiang Yue, Yu Su, and Huan Sun. Grokking of implicit reasoning in transformers: A mechanistic journey to the edge of generalization. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/ad217e0c7fecc71bdf48660ad6714b07-Abstract-Conference.html.

Ke Alexander Wang, Jiaxin Shi, and Emily B. Fox. Test-time regression: a unifying framework for designing sequence models with associative memory. *arXiv:2501.12352*, 2025. URL https://arxiv.org/abs/2501.12352.

Kaiyue Wen, Xingyu Dang, and Kaifeng Lyu. RNNs are not transformers (yet): The key bottleneck on in-context retrieval. *arXiv:2402.18510*, 2024. URL https://arxiv.org/abs/2402.18510.

Jennifer C. White and Ryan Cotterell. Examining the inductive bias of neural language models with artificial languages. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long*

Papers), pages 454–463, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.38. URL https://aclanthology.org/2021.acl-long.38/.

Sarah Wiegreffe, Oyvind Tafjord, Yonatan Belinkov, Hannaneh Hajishirzi, and Ashish Sabharwal. Answer, assemble, ace: Understanding how LMs answer multiple choice questions. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=6NNA0MxhCH.

Zhengxuan Wu, Atticus Geiger, Aryaman Arora, Jing Huang, Zheng Wang, Noah Goodman, Christopher Manning, and Christopher Potts. pyvene: A library for understanding and improving PyTorch models via interventions. In Kai-Wei Chang, Annie Lee, and Nazneen Rajani, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: System Demonstrations)*, pages 158–165, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-demo.16. URL https://aclanthology.org/2024.naacl-demo.16/.

Kayo Yin and Jacob Steinhardt. Which attention heads matter for in-context learning? *arXiv:2502.14010*, 2025. URL https://arxiv.org/abs/2502.14010.

Wei Zhang and Bowen Zhou. Learning to update auto-associative memory in recurrent neural networks for improving sequence memorization. *arXiv:1709.06493*, 2017. URL https://arxiv.org/abs/1709.06493.

Appendix

Table of Contents

A Formal definitions and details for ATR

For reference, we provide formal definitions for PCFGs and the normal form we use in ATR.[7]

Definition A.1. A **probabilistic context-free grammar** is a tuple $\mathcal{G} = \langle \mathcal{N}, \Sigma, \mathrm{S}, \mathcal{R}, p \rangle$ where:

- $\mathcal{N}$ is a finite set of non-terminal symbols;
- Σ is an alphabet of terminal symbols;
- $\mathrm{S} \in \mathcal{N}$ is a start symbol;
- $\mathcal{R} \subset \mathcal{N} \times (\mathcal{N} \cup \Sigma)^*$ is a finite set of production rules, mapping a left-hand side symbol $\mathrm{N} \in \mathcal{N}$ to a string of symbols that may be either terminals or nonterminals; each such rule is written as $\mathrm{X} \to \boldsymbol{\alpha}$;
- $p : \mathcal{R} \to [0, 1]$ is a weighting function which assigns a probability to each production rule for a nonterminal; this function is locally normalised, meaning $\{\sum_{\mathrm{X} \to \boldsymbol{\alpha}} p(\mathrm{X} \to \boldsymbol{\alpha}) = 1 \mid \mathrm{X} \in \mathcal{N}\}$.

Definition A.2. A PCFG $\mathcal{G} = \langle \mathcal{N}, \Sigma, \mathrm{S}, \mathcal{R}, p \rangle$ is in **Greibach normal form (GNF)** if each production rule in $\mathcal{R}$ is of the form $\mathrm{X} \to a\ \mathrm{X}_1\ \ldots\ \mathrm{X}_\mathrm{n}$, where $\mathrm{X}_1, \ldots, \mathrm{X}_n \in \mathcal{N}$ and n may be 0. Similarly, a PCFG is in **right-Greibach normal form** if each rule is of the form $\mathrm{X} \to \mathrm{X}_1\ \ldots\ \mathrm{X}_\mathrm{n}\ a$.

For ATR, the PCFG is in Greibach normal form if the head is the leftmost symbol of the production rule's righthand side; similarly, if the PCFG is right-headed, it is in right-Greibach normal form.

Definition A.3. A **derivation step** $\boldsymbol{\alpha} \Rightarrow \boldsymbol{\beta}$ is an operation where, given strings of symbols $\boldsymbol{\alpha}, \boldsymbol{\beta} \in (\mathcal{N} \cup \Sigma)^*$, the leftmost nonterminal $\mathrm{X} \in \mathcal{N}$ in $\boldsymbol{\alpha}$ is rewritten using the right-hand side of a production rule $\mathrm{X} \to \ldots \in \mathcal{R}$ to obtain $\boldsymbol{\beta}$.

Definition A.4. A **derivation** under the PCFG $\mathcal{G}$ is a sequence of strings $[\boldsymbol{\alpha}_0, \ldots, \boldsymbol{\alpha}_m]$ where $\boldsymbol{\alpha}_0 \in \mathcal{N}$ and each step $\boldsymbol{\alpha}_{i+1}$ is formed by a derivation step on $\boldsymbol{\alpha}_i$. The final string $\boldsymbol{\alpha}_m \in \Sigma^*$ is the **yield** of the derivation.

Each ATR document is the yield of a derivation sampled under the GNF PCFG $\mathcal{G}$.

A.1 Additional details on ATR

Parent terminals in GNF. We set the left/right-most terminal in each production rule (which leads to the GNF property) the parent of all other generated terminals. This terminal is sampled specially: for each nonterminal, we independently sample a distribution over terminals from a uniform Dirichlet, and for all production rules with that nonterminal on the lefthand side we use that distribution to sample the parent terminal. This simulates how heads of phrases in natural language (analogous to our parent terminals) decide the type of the phrase they head (analogous to our nonterminals).

Maximum depth. To enforce maximum depth, we first assign a uniformly random depth score $d : \mathcal{N} \to \mathbb{N} \in \{1, \ldots, \texttt{max\_depth}\}$ to each nonterminal in the vocabulary. Then, for each production rule for each nonterminal X, we only allow nonterminals Y with $d(\mathrm{Y}) > d(\mathrm{X})$ on the right-hand side. Note that this means no recursion is possible.

[7] We use similar formalisations of PCFGs as previous work in NLP, e.g. Nowak and Cotterell [2023].

B Model configurations

Table 2: Default model configurations across all architectures. In experiments, we sweep learning rate and embedding dimension, reporting results from the instance with highest accuracy.

(a) Attention

Parameter	Values
dropout	0.0
num_heads	1

(b) Hyena

Parameter	Values
l_max	1024
filter_order	64
num_heads	1
num_blocks	1
outer_mixing	False
dropout	0.0
filter_dropout	0.0
short_filter_order	3
bidirectional	False

(c) BaseConv

Parameter	Values
l_max	1024
kernel_size	$[3, -1]$
implicit_long_conv	True
use_act	False

(d) Based

Parameter	Values
BaseConv	
l_max	1024
kernel_size	3
implicit_long_conv	True
use_act	False
Based	
l_max	1024
feature_dim	8
num_key_value_heads	1
num_heads	1
feature_name	taylor_exp
train_view	quadratic

(e) H3

Parameter	Values
l_max	1024
d_state	1024
head_dim	1024

(f) Mamba

Parameter	Values
d_conv	4

C Task hyperparameters

Table 3: Task hyperparameters.

(a) Parameters used for constructing AR documents.

Parameter	Values
L_{max}	32
L_{min}	32
$\lvert\Sigma\rvert$	$\{8192\}$

(b) Parameters used for constructing ATR documents.

Parameter	Values
H	Right
d_{max}	10
L_{max}	$\{5, 10\}$
R_{max}	5
$\lvert\mathcal{N}\rvert$	40
$\lvert\Sigma\rvert$	$\{20, 8192\}$
r_{Σ}	20

D More experiments on AR and ATR

Many parameters of synthetic tasks like AR and ATR and the model architectures we tested have interesting effects on behavioural and mechanistic metrics, but not all experiments could fit in our main text. Therefore, we include additional interesting observations in this appendix.

D.1 Attention needs position embeddings

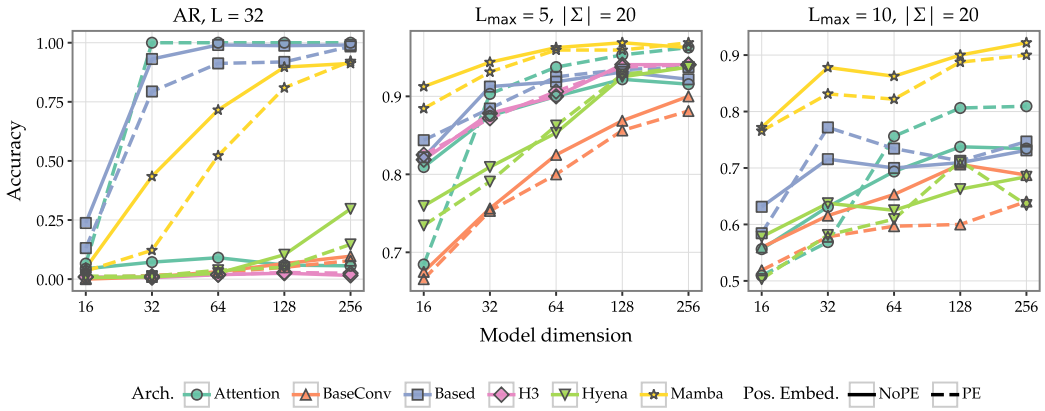
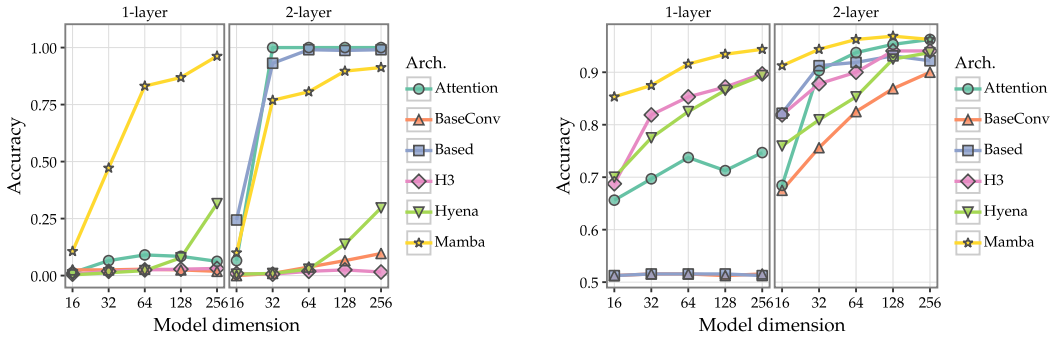


Figure 7: **Position embedding**: Model accuracy on AR and two ATR settings with and without absolute position embeddings.

Due to an initial configuration mistake, we accidentally trained all architectures with absolute position embeddings; in the `zoology` codebase [Arora et al., 2024a], only Attention is meant to be trained in this way. Fortuitously, this resulted in an interesting ablation: do SSMs, which are usually trained without it, also benefit from position embeddings?

Behavioural results. Our results in Figure 7 resoundingly show no: SSMs generally perform worse with position embeddings (PE). Attention is highly dependent on PE; performance on AR drops from 100.00% to 5.62% at $d = 256$ with NoPE. Attention lacks recurrence, unlike SSMs, so this is not surprising. However, on ATR, at smaller dimensionalities NoPE actually outperforms PE Attention. Further ablations ought to consider alternative PE methods such as RoPE and Alibi.

D.2 1-layer SSMs learn direct retrieval on AR and ATR



(a) Accuracy of 1-layer vs. 2-layer models on AR, 32 key–value pairs. 1-layer induction models fail.

(b) Accuracy of 1-layer vs. 2-layer models on ATR ($L = 5, |\Sigma| = 20$), with Based and BaseConv failing.

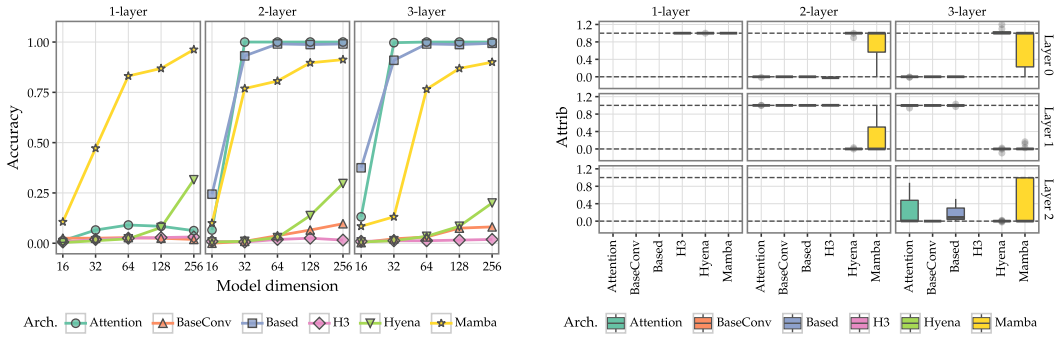
Figure 8: **1-layer models on AR and ATR**: Architectures that learn induction in the 2-layer setting fail to perform non-trivially with 1 layer. Mamba is highly performant with 1 layer on both tasks.

Throughout our experiments on AR and ATR, we have claimed that SSMs (except for Based and possibly BaseConv) learn a direct retrieval mechanism which does not require an intermediate step

like attention, i.e. only a single SSM layer is needed to learn AR and ATR. To verify this, we repeat AR and $L = 5, |\Sigma| = 20$ ATR experiments (without train–test split) with 1-layer models.

Behavioural results. We find comparable performance for direct retrieval models between 1-layer and 2-layer settings on AR (Figure 8a). In fact, at $d = 256$, 1-layer Mamba (96.25%) outperforms 2-layer Mamba (91.25%), as does Hyena (31.56% vs. 29.69%). 1-layer Based and BaseConv are architecturally identical, so we only report one; that architecture and Attention, both relying on induction in the 2-layer case, fail to learn AR with one layer. On ATR (Figure 8b), we see a more noticeable difference with layer count on all architectures, but again Attention, Based, and BaseConv become the worst architectures with one layer (e.g. $96.25\% \rightarrow 74.69\%$ for Attention at $d = 256$).

D.3 SSMs prefer layer 0 to perform AR



(a) Accuracy with 1–3 layers on AR.

(b) Attribution scores with 1–3 layers on AR.

Figure 9: **Varying layer count on AR**: Behavioural and mechanistic evaluations for models with 1–3 layers on AR.

Since we have confirmed that the direct retrieval mechanism in SSMs requires only a single layer, we are curious which layer this mechanism forms in if more than two layers are present. We train models with up to three layers on AR and report results.

Behavioural results. 3-layer models perform about the same on AR as 2-layer models across architectures (Figure 9a), except for a large drop in performance for Mamba when $d = 32$; this may just be an optimisation failure.

Mechanistic analysis. For our mechanistic metric, instead of intervening on each block, we intervene at the sequence mixer's output to the *query* token in each layer; this tells us if that layer is directly responsible for writing the answer to the output position. We apply the same filter as in §5.1, with a threshold of 0.01. Figure 9b shows that among performant models, Hyena and Mamba prefer layer 0 for performing AR no matter the layer count; however, some Mamba checkpoints learn the mechanism in the final layer as well (but never layer 1 in a 3-layer model). Attention, Based, and BaseConv prefer layer 1, which is expected since this is the second step of the induction mechanism. However, some checkpoints of Attention and Based also have non-zero attribution score at layer 2 in the 3-layer setting.

D.4 Rightmost sibling queries are trivial for all architectures

Since ATR has hierarchical structure, we attempted an initial experiment with multihop queries; specifically, we present queries where the answer is that terminal's rightmost sibling terminal. Models are only trained on this type of query, not standard parent queries as reported in the main text. We train with the same settings in §3.3.

Behavioural results. In Figure 10 we show that all models (except Based and BasedConv with 1 layer, where they only have local convolutions) achieve greater than 80% accuracy at the task at all dimensionalities. We see slight improvement from 1-layer to 2-layer models but at this point performance is saturated and 3-layer does not help. Clearly, this task is extremely simple for all models, even more so than parent queries, and thus does not provide useful signal for comparing architectures.

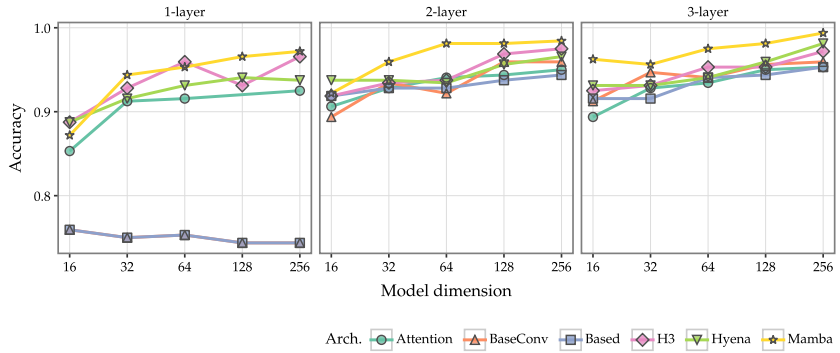


Figure 10: **Sibling queries**: Accuracy across models with 1–3 layers on ATR ($L_{\text{max}} = 5, |\Sigma| = 20$)
.

Why are sibling queries easy? Parent nodes are guaranteed to be special terminals in our GNF which are sampled from a nonterminal-dependent distribution (see appendix A). However, siblings have a large chance of being fixed terminals specified by the production rule. Additionally, the rightmost sibling of a particular terminal may be itself, if it is the rightmost terminal of its production rule. We speculate that these factors combined make sibling queries easier than parent queries, and thus not a suitable testbed for multihop reasoning.

Future work. The appropriate analogue to study multihop *reasoning* in ATR is grandparent relations (or higher up ancestors in the tree), since the grandparent is always a special head terminal (like the parent) and is always to the right of the parent and thus different from the query terminal. We leave further experiments on this to future work.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We make specific and explicit claims about the performance of SSMs and attention on both synthetic tasks against our mechanistic evaluations.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We address limitations of interpretability-based evaluation in the Discussion section (§7), and point out directions for future work.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [NA] .

Justification: Our paper contains no theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: We provide a detailed description of our setup in §3 and §4, as well as a formalization of our PCFG setup in Appendix A. We will release code for the camera-ready; in addition, we include links to the `Zoology` and `pyvene` repositories which contain much of the infrastructure we used for running experiments.

Guidelines:

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

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

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

   Answer: [Yes]

   Justification: We make anonymized code available as part of the supplemental information.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: See §3.3, as well as §5 and appendices B and C.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We include error bars on all aggregated plots.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: §3.3 includes a summary of necessary compute details.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We did not use human participants. All data was synthetically generated. Our work has no forseeable harmful consequences.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [NA]

    Justification: Our work focuses on evaluating small-scale architecture research and thus has no immediate societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

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

11. **Safeguards**

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

    Answer: [NA]

    Justification: Our models are trained on synthetic tasks, and our data is synthetically generated; we thus do not foresee any risk of misuse.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We cite and include links to both `Zoology` [Arora et al., 2024a] and `pyvene` [Wu et al., 2024], both of which our codebase is built upon.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: Documentation of how to use our codebase is provided in a `README.md` file.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: Our work does not use human subjects.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: Our work does not use human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [NA]

    Justification: We did not use LLMs as part of this work.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.