

Succinctness, Not Expressivity, Predicts Syntactic Competence

Shangao Li^{1,2*} Jinghan Wu^{1,3*} Chang Lu⁴ Yizheng Zhao^{1,3†}

¹State Key Laboratory of Novel Software Technology, Nanjing University, Nanjing, China

²School of Computer Science, Nanjing University, Nanjing, China

³School of Artificial Intelligence, Nanjing University, Nanjing, China

⁴Computational Biology and Biomedical Informatics Program,
Yale University, New Haven, CT, USA

Abstract

Transformers outperform state-space models and recurrent architectures on long-distance syntactic dependencies, yet no theory predicts which architecture will win on which task. For syntactic phenomena expressible as short B-RASP programs, Transformers need a fixed number of layers regardless of dependency distance, while sequential models face growing effective complexity. We derive a *succinctness prediction table* ranking Transformers, LSTMs, and Mamba on six English syntactic phenomena before any experiment, with GRU and RWKV as additional sequential controls. All predictions are confirmed: Transformer accuracy exceeds 96% at maximum structural complexity while Mamba drops below 55%, a gap that persists at $79\times$ the Transformer’s parameter count. To verify the winning architecture succeeds *for the predicted reason*, Distributed Alignment Search (DAS), a causal intervention method, finds alignment of 0.95+ in Transformers but near the 0.50 chance baseline in Mamba, validated by seven families of negative controls. Training dynamics reveal that Transformers assemble the predicted algorithm in causal-dependency order, whereas Mamba achieves high accuracy without it, producing an *accuracy-alignment gap* invisible to behavioral benchmarks. Validation on Pythia confirms these patterns transfer to pre-trained models.

1 Introduction

In *The keys to the cabinet are on the table*, the verb *are* agrees with the distant plural subject *keys*, not the adjacent singular attractor *cabinet*; correctly predicting this requires a language model to identify the agreement controller across intervening material. Linzen et al. (2016) introduced this *number agreement* task as a controlled probe of syntactic competence; subsequent work extended

the paradigm to reflexive binding, negative polarity item (NPI) licensing, and dozens of other phenomena (Marvin and Linzen, 2018; Gulordava et al., 2018), yielding the 67-paradigm BLiMP benchmark (Warstadt et al., 2020). One finding is consistent across these evaluations: Transformers (Vaswani et al., 2017) outperform recurrent models on long-distance dependencies, and the gap widens with the number of intervening distractors (Hu et al., 2020; Misra and Mahowald, 2024). Selective state-space models (SSMs) such as Mamba (Gu and Dao, 2023) match Transformers on perplexity and aggregate benchmarks (Waleffe et al., 2024), yet still fail on retrieval of a distant token’s features (Wen et al., 2025). The gap is well documented, but no existing theory predicts which architecture will succeed on which syntactic task.

The most natural explanation is *expressivity*: perhaps recurrent models cannot represent the required computation. Formal analysis rules this out. Transformers and SSMs both lie in TC^0 (constant-depth threshold circuits; Merrill et al., 2024); LSTMs (Hochreiter and Schmidhuber, 1997) are strictly more powerful than simple RNNs (Merrill et al., 2020; Delétang et al., 2023); and bounded-distance agreement requires only a three-state deterministic finite automaton (DFA) that any of these architectures can implement. Delétang et al. (2023) map architectures to Chomsky-hierarchy boundaries; we address an orthogonal question: within a class that all studied architectures can express, what is the encoding-cost gap? Shortcut learning (Geirhos et al., 2020) is also insufficient: all architectures exploit surface heuristics on easy inputs, but only Transformers maintain accuracy as distance grows (Bhattamishra et al., 2023; Sarrof et al., 2024). Since all architectures can express the same computation and share the same training signal, the gap must reflect a difference in how *compactly* each architecture can represent the solution.

The RASP programming language (Weiss et al.,

* Equal contribution; listed alphabetically by last name.

† Corresponding author: zhaoyz@nju.edu.cn

2021) formalizes Transformer computations as symbolic programs; its Boolean fragment B-RASP (Yang et al., 2024) captures exactly the star-free regular languages, equivalent to first-order logic with linear order ($\text{FO}[\prec]$) and to counter-free automata (McNaughton and Papert, 1971; Barceló et al., 2024; Strobl et al., 2024). Bergsträßer et al. (2026) prove that hard-attention Transformers are exponentially more succinct than linear temporal logic (LTL) formulas and fixed-precision recurrent networks (hence, by extension, state-space models), and doubly exponentially more succinct than DFAs recognizing the same languages (§2.2). In concrete terms: a B-RASP program of k operations maps to a k -layer Transformer with $O(d_h^2)$ parameters per layer, independent of input length or dependency distance; a sequential model (a recurrent network or SSM, processing the input through a fixed-size state) may need state dimension that grows exponentially with the number of distractors. Decompilation work (Huang et al., 2026, 2025) corroborates these formal predictions empirically.

These separations translate into testable predictions for natural language syntax (Figure 1). A phenomenon that admits a two-operation B-RASP program (one attention-based retrieval, one Boolean check) should be solved by a two-layer Transformer at any dependency distance, while a recurrent or state-space model should degrade with structural complexity; for local phenomena where a minimal automaton suffices, no gap should appear. We formalize this into a *succinctness prediction table* covering six phenomena across three architectures, derived entirely from B-RASP program analysis before any experiment (§3). Throughout, *syntactic competence* refers to these six controlled phenomena within the star-free fragment of §2.2, operationalized over template-generated sequences, not to natural-language syntax in full generality.

Behavioral confirmation alone, however, cannot distinguish a model that succeeds for the predicted reason from one that achieves the same accuracy through an unrelated mechanism. Probing studies illustrate the problem: representations may encode syntactic features without the model using them as predicted (Belinkov and Glass, 2019; Hewitt and Manning, 2019; Clark et al., 2019). Causal abstraction (Geiger et al., 2021, 2025; Pearl, 2009) addresses this gap by testing whether a high-level causal model faithfully describes a network’s computation through interventions, not correlations. Distributed Alignment Search (DAS; Geiger et al.,

2024) operationalizes the framework by learning orthogonal rotations that maximize interchange intervention accuracy (IIA) with respect to hypothesized causal variables. Known concerns about alignment specificity (Sutter et al., 2025), dormant pathway activation (Makelov et al., 2024), and causal explanation multiplicity (Méloux et al., 2025) motivate our use of seven families of negative controls, to our knowledge the broadest suite in the causal-abstraction literature. Tracking IIA across training connects to progress measures for grokking (Nanda et al., 2023; Li et al., 2026).

Our work yields four insights:

- (1) *Succinctness, not expressivity, is the right lens for architecture comparison on syntax.* A prediction table derived only from B-RASP program analysis correctly predicts the parallel-sequential partition and the distance scaling for five architectures on all phenomena (§3–5.1).
- (2) *The gap is structural and resists scaling.* Mamba at $79\times$ the Transformer’s parameter count still cannot close the gap; the Transformer saturates by $d_h = 64$ while Mamba shows no saturation at $d_h = 1024$ (§5.1).
- (3) *The winning architecture succeeds for the predicted reason.* In Mamba, retrieval partially degrades (atomic IIA ≈ 0.62) and downstream composition collapses (compositional IIA $\approx$ chance), matching the predicted bottleneck; Transformers implement all eight variables faithfully (IIA ≥ 0.95 ; §5.2).
- (4) *High behavioral accuracy can mask algorithmic failure.* Mamba reaches 97% on easy instances without the prescribed algorithm, an *accuracy-alignment gap* invisible to behavioral benchmarks; validation on Pythia confirms the pattern transfers to pre-trained models (§5.3).

2 Preliminaries

2.1 B-RASP and Star-Free Languages

RASP (Weiss et al., 2021) is a programming language whose operations correspond one-to-one to the components of a Transformer layer. Its Boolean fragment, B-RASP (Yang et al., 2024), restricts all intermediate values to $\{0, 1\}$ and operates on an input sequence $x = (x_1, \dots, x_n)$ over a finite alphabet Σ through three types of operations. *Input indicators*: for each symbol $\sigma \in \Sigma$, the Boolean sequence Q_σ is defined by $Q_\sigma(i) = \mathbb{1}[x_i = \sigma]$. *Hard-attention selection*: given Boolean sequences ϕ (selector) and ψ (value), $\text{HA}_\leftarrow[\phi; \psi](i)$ (attend-

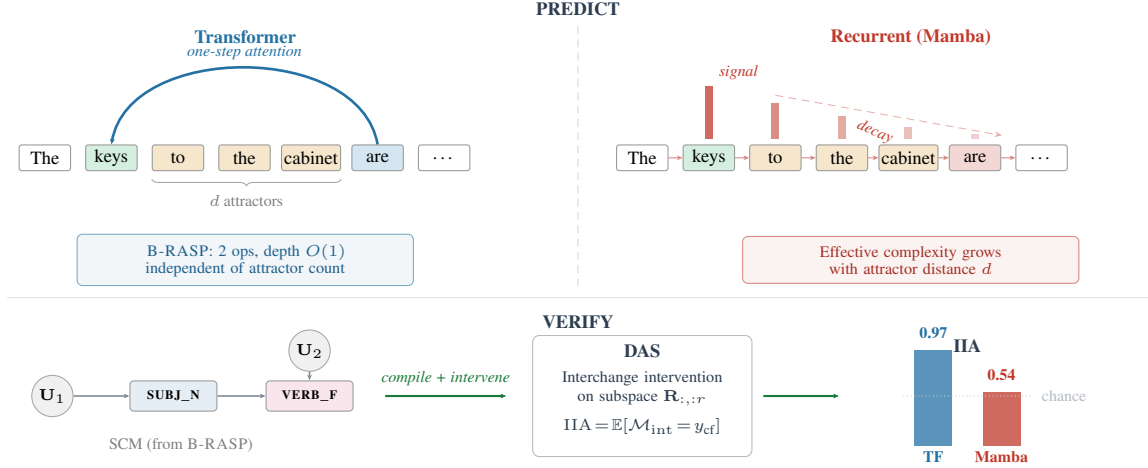


Figure 1: Overview illustrated on subject-verb agreement with attractors. **Top (Predict):** A B-RASP program retrieves the subject’s number in one attention step regardless of attractor count, yielding $O(1)$ depth for Transformers. A recurrent model must propagate state through each attractor, facing signal decay that grows with distance d . **Bottom (Verify):** The B-RASP program compiles to a two-variable structural causal model (SCM). DAS locates these variables in the trained Transformer (IIA = 0.97) but not in Mamba (IIA = 0.54).

ing leftward) returns $\psi(j)$ where j is the *leftmost* position with $j \leq i$ and $\phi(j) = 1$; rightward and bidirectional variants are defined analogously, and each such operation corresponds to a single attention head. *Pointwise Boolean:* any function $f: \{0, 1\}^m \rightarrow \{0, 1\}$ may be applied position-wise to existing Boolean sequences, yielding a new sequence $s(i) = f(s_1(i), \dots, s_m(i))$; this corresponds to a feedforward (MLP) sublayer.

The *size* of a B-RASP program P , written $|P|$, is the number of hard-attention and pointwise operations (input indicators not counted). A program of size k can be implemented by a k -layer masked hard-attention Transformer (Yang et al., 2024).

The central characterization theorem establishes that the following are equivalent (Yang et al., 2024; McNaughton and Papert, 1971): (i) languages recognizable by masked hard-attention Transformers, (ii) languages definable by B-RASP programs, (iii) star-free regular languages, (iv) languages definable in first-order logic with linear order ($\text{FO}[\prec]$), and (v) languages recognized by counter-free automata. See Strobl et al. (2024) for a comprehensive survey. The C-RASP extension (Yang et al., 2025) adds counting and establishes a strict depth hierarchy beyond the star-free fragment; all syntactic phenomena we study fall within the B-RASP (star-free) scope.

2.2 Succinctness

These formalisms recognize exactly the same class of languages; they differ in how concisely they can

describe a given language within it. A model class $\mathcal{F}_1$ is *exponentially more succinct* than $\mathcal{F}_2$ if there exists a family of languages $\{L_n\}_{n \geq 1}$ such that every $\mathcal{F}_1$ description of L_n has size $O(\text{poly}(n))$ while every $\mathcal{F}_2$ description requires size $\Omega(2^n)$.

Bergsträßer et al. (2026) prove that hard-attention Transformers (equivalently, B-RASP programs) are exponentially more succinct than LTL formulas, doubly exponentially more succinct than deterministic finite automata (DFAs), and, most directly relevant here, exponentially more succinct than fixed-precision recurrent networks and, by extension, state-space models: there exist star-free languages for which a B-RASP program of size $O(n)$ requires an LTL formula of size $\Omega(2^n)$, a DFA with $\Omega(2^{2^n})$ states, and a fixed-precision RNN of size $\Omega(2^n)$. These separations are stated for a language family indexed by n ; our phenomena instantiate the fixed program at $n = 1$, and the separation enters through the family indexed by dependency distance d , along which the sequential state requirement grows while the B-RASP program is unchanged (Appendix B.2).

For the syntactic phenomena we study, the implication is direct. A constraint such as subject-verb agreement admits a B-RASP program of constant size ($|P| = 2$), regardless of how many attractors intervene between subject and verb. A Transformer implementing this program needs only two layers, with a per-layer parameter count that does not depend on input length. A sequential model, by contrast, must propagate the subject’s number feature

through each intervening token via recurrent state updates. In selective state-space models (Gu and Dao, 2023), the state transition at position t takes the form $\mathbf{h}_t = \mathbf{A}_t \mathbf{h}_{t-1} + \mathbf{B}_t x_t$, where the matrices $\mathbf{A}_t$ and $\mathbf{B}_t$ depend on the input. Maintaining a specific signal (e.g., subject number) across d input-dependent transitions requires that $\prod_{t=1}^d \mathbf{A}_t$ approximately preserves the corresponding direction, a condition that becomes exponentially less likely as d grows for generic learned parameters (Merrill et al., 2024; Wen et al., 2025).

2.3 Causal Verification with DAS

Behavioral accuracy confirms *what* a model computes; causal abstraction (Geiger et al., 2021, 2025) tests *how* it computes it. The framework asks whether a high-level causal model faithfully summarizes a neural network’s internal processing.

A structural causal model (SCM; Pearl, 2009) $\mathcal{C} = (\mathbf{U}, \mathbf{V}, \mathbf{F})$ consists of exogenous (input) variables $\mathbf{U}$, endogenous variables $\mathbf{V} = \{V_1, \dots, V_m\}$, and deterministic functions $\mathbf{F} = \{f_1, \dots, f_m\}$ with $V_j = f_j(\text{Pa}(V_j))$, where $\text{Pa}(V_j) \subseteq \mathbf{U} \cup \mathbf{V} \setminus \{V_j\}$ denotes the parents of V_j . In our setting, each B-RASP operation compiles to one endogenous variable: a hard-attention step yields an *atomic* variable that depends only on exogenous inputs (e.g., SUBJ_NUM, which retrieves the subject’s grammatical number), and a pointwise Boolean step yields a *compositional* variable that depends on the atomic variable and a local input (e.g., VERB_FORM, which checks agreement between subject number and verb inflection). The full compilation procedure is in §3 and Appendix A. The SCM is architecture-neutral: its variables are properties of the task’s causal structure, and DAS searches for them in any network, with no assumption about attention, gating, or recurrence.

Scope of the theoretical claim. The succinctness theorems are proved for hard attention. No formal bridge from these theorems to trained softmax models is currently available; characterizing softmax-attention expressivity remains an open problem (Strobl et al., 2024), although parameter-uniform, temperature-scaled soft-attention simulations now cover every masked unique-hard-attention Transformer via LTL (Yang et al., 2026). We use the theorems to generate directional, falsifiable predictions; their verification in §5.2 is empirical and causal, not a corollary of the theorems.

Distributed Alignment Search (DAS; Geiger

et al., 2024) tests whether a hypothesized variable V is encoded in a model’s representations; it scales to billion-parameter models (Wu et al., 2023). Given a model $\mathcal{M}$ with hidden representation $\mathbf{h}^\ell \in \mathbb{R}^{d_h}$ at layer ℓ , DAS learns an orthogonal matrix $\mathbf{R} \in O(d_h)$ via Cayley parameterization and designates its first r columns as a rank- r subspace. An *interchange intervention* replaces, in this subspace, the base input’s activations with those from a source input while leaving the complementary subspace intact:

$$\mathbf{h}_{\text{int}} = \mathbf{h}_b + \mathbf{R}_{:,r}(\mathbf{R}_{:,r}^\top \mathbf{h}_s - \mathbf{R}_{:,r}^\top \mathbf{h}_b), \quad (1)$$

where $\mathbf{h}_b$ and $\mathbf{h}_s$ are the base and source representations at layer ℓ , and $\mathbf{R}_{:,r}$ denotes the first r columns of $\mathbf{R}$. The model then runs forward from layer ℓ using $\mathbf{h}_{\text{int}}$ in place of $\mathbf{h}_b$.

The quality of alignment is measured by interchange intervention accuracy (IIA). Given counterfactual pairs $\mathcal{D}_{cf} = \{(x_b, x_s, y_{cf})\}$, where y_{cf} is the SCM-predicted output when V takes its value from x_s while all other exogenous variables retain their x_b values:

$$\text{IIA}(V) = \frac{1}{|\mathcal{D}_{cf}|} \sum_{(x_b, x_s, y_{cf}) \in \mathcal{D}_{cf}} \mathbb{1}[\mathcal{M}_{\text{int}}(x_b, x_s) = y_{cf}],$$

where $\mathcal{M}_{\text{int}}(x_b, x_s)$ denotes the model’s output after applying the interchange intervention of Equation (1) with the learned $\mathbf{R}$, r , and ℓ . An IIA of 1.0 indicates perfect causal alignment; 0.5 is chance for binary variables. DAS optimizes $\mathbf{R}$ via gradient descent to maximize IIA on a training split; configuration selection and reporting follow the validation/test protocol of §4.

DAS is the top-performing method on the Mechanistic Interpretability Benchmark (MIB; Mueller et al., 2025) for causal variable localization. To guard against known failure modes, we restrict alignment to low-rank orthogonal rotations rather than unconstrained nonlinear maps, which can yield high IIA even in randomly initialized networks (Sutter et al., 2025); we intervene on residual-stream representations rather than MLP internals, avoiding the dormant-pathway artifacts of Makelov et al. (2024); and we do not claim that the B-RASP-derived SCM is the unique explanation of model behavior, addressing the non-identifiability concern of Méloux et al. (2025). Seven families of negative controls (§4) provide empirical calibration.

3 Succinctness Predicts Architecture

3.1 From Syntax to B-RASP Programs

We show that six syntactic phenomena can each be expressed as a two-operation B-RASP program: one hard-attention step that retrieves a grammatical feature from a distant position (atomic), followed by one pointwise Boolean step that checks compatibility with a local feature (compositional). Each program compiles directly to a two-variable SCM.

Subject-verb agreement. Consider *The keys to the cabinet are on the table*. The singular noun *cabinet* is an *attractor*: it intervenes between the plural subject *keys* and the verb *are*, and the number of such intervening nouns is the *attractor distance*, denoted d . We define the derived indicator $Q_{\text{noun}} = Q_{\text{n_sg}} \vee Q_{\text{n_pl}}$, which marks any noun position. The B-RASP program is:

$$\text{SUBJ_NUM} = \text{HA}_{\leftarrow}[Q_{\text{noun}}; Q_{\text{n_pl}}] \quad (2)$$

$$\text{VERB_FORM} = \text{SUBJ_NUM} \leftrightarrow Q_{\text{v_pl}} \quad (3)$$

Step (2) attends to the leftmost noun and returns whether it is plural ($\text{SUBJ_NUM} = 1$) or singular ($= 0$). Step (3) checks whether the subject’s number matches the verb’s inflection via the bi-conditional. The compiled SCM has two endogenous variables linked by $\text{U} \rightarrow \text{SUBJ_NUM} \rightarrow \text{VERB_FORM}$, with VERB_FORM additionally depending on the input feature $Q_{\text{v_pl}}$.

NPI licensing. A negative polarity item (NPI) such as *ever* requires a c-commanding licenser such as *no* or *never*. We define:

$$\text{HAS_LIC} = \text{HA}_{\leftarrow}[Q_{\text{lic}}; Q_{\text{lic}}] \quad (4)$$

$$\text{NPI_OK} = \neg Q_{\text{npi}} \vee \text{HAS_LIC} \quad (5)$$

Step (4) attends to the leftmost licenser and returns 1 if one exists, 0 otherwise. Step (5) accepts either non-NPI positions or NPI positions that are licensed.

Remaining phenomena. Reflexive binding (ANTE_NUM, REFL_OK), determiner-noun concord (DET_NUM, CONC_OK), multi-clause NPI, and cross-clausal binding all follow the same two-step template; their programs are given in Appendix A. The key observation is that $|P| = 2$ for every phenomenon, and this value is *invariant* to the dependency distance d : adding more attractors or clause boundaries does not add B-RASP operations. It is this invariance that generates the predictions in §3.2.

3.2 Architecture Predictions

Transformers. The constant program size ($|P| = 2$) has asymmetric consequences for the two architecture families. By the construction of §2.1 (one attention head per hard-attention operation, one feedforward sublayer per pointwise operation), a program of size k is implementable by a k -layer Transformer with $O(d_h^2)$ parameters per layer, independent of input length n and dependency distance d . For all six phenomena, $k = 2$.

Sequential models. For sequential models, each attractor forces an input-dependent state transition that can corrupt the stored signal (§2.2); the effective state complexity therefore grows with d , while B-RASP depth remains $O(1)$. For local phenomena such as determiner-noun concord, where $d \leq 1$ by construction, this gap vanishes.

Predictions. Table 1 collects the resulting predictions. Predictions are *directional* (which architecture wins and how the gap scales with d), not numerical (the exact gap at a given d depends on training details the theory does not specify). Every prediction is falsifiable: if Mamba accuracy remained flat across increasing d , or if the Transformer degraded, the succinctness account would be refuted.

Table 1: Succinctness predictions. $|P|$: required Transformer depth (constant). Sequential bottleneck: source of state-propagation difficulty. $\uparrow d$: gap grows with attractor distance; $\uparrow \text{depth}$: gap grows with clause depth.

Phenomenon	$ P $	Seq. bottleneck	Predicted gap
Concord (local)	2	minimal	None
Agr., $d=0$	2	minimal	Weak
Agr., d attractors	2	interf. $\propto d$	Strong, $\uparrow d$
NPI licensing	2	interf. $\propto d$	Strong, $\uparrow d$
Reflexive bind.	2	interf. $\propto d$	Strong, $\uparrow d$
Multi-clause NPI	2	$\propto$ clause depth	Strong, $\uparrow \text{depth}$
Cross-cl. binding	2	interf. $\propto d$	Strong, $\uparrow d$

4 Experimental Setup

Tasks and data. We evaluate the six phenomena of §3.1: subject-verb agreement, NPI licensing, reflexive binding, determiner-noun concord, multi-clause NPI, and cross-clausal binding. For each phenomenon, template-based sentences are generated from context-free grammars in the style of Marvin and Linzen (2018), using the 11-symbol alphabet of Appendix A. Attractor distance is varied over $d \in \{0, 1, 2, 4, 6, 8\}$; clause nesting depth

over $\{1, 2, 3\}$ for the two multi-clause variants. Counterfactual pairs (x_b, x_s, y_{cf}) for DAS are constructed so that x_b and x_s differ in exactly one prescribed variable, with y_{cf} set by the compiled SCM (§3.1). Each variable’s DAS evaluation uses 10K training and 2K test counterfactual pairs (8K/1.6K for concord due to limited distance range); behavioral datasets are larger (up to 50K training sentences) and detailed in Appendix C. Pre-trained models are evaluated on 23 BLiMP paradigms (Warstadt et al., 2020) covering agreement and NPI. Full generation details appear in Appendix C.

Models. From-scratch models are two-layer architectures with hidden dimension $d_h = 128$: a Transformer (401K params) with 4 attention heads; an LSTM (269K; Hochreiter and Schmidhuber, 1997); and a Mamba model (239K; Gu and Dao, 2023) with SSM state dimension $N = 16$. The Transformer’s parameter advantage ($1.68\times$ over Mamba) makes this a conservative comparison; the width-scaling experiment in §5.1 shows that the gap persists even when Mamba has $79\times$ more parameters. GRU and RWKV-4 (Peng et al., 2023) baselines with matched depth and comparable parameters are included in Appendix E. All from-scratch models share one objective: a linear classification head on the final token’s hidden state, two classes, cross-entropy loss; supervision is identical across architectures and phenomena (Appendix C). All are trained with AdamW (Loshchilov and Hutter, 2019) (learning rate 10^{-3} , weight decay $\lambda = 1.0$, batch size 512) for 40K steps across 5 random seeds. Robustness of the results to the training configuration is assessed in Appendix I (optimizer and DAS hyperparameters), Appendix I.4 (weight decay, including a sweep over λ), and Appendix I.8 (SSM state dimension). The high weight decay follows Power et al. (2022) to induce grokking dynamics. For width scaling, d_h is varied over $\{32, 64, 128, 256, 512, 1024\}$ for all three main architectures while other hyperparameters are held fixed, yielding parameter counts from ~ 77 K to ~ 17 M. Pre-trained evaluation uses Pythia 160M, 410M, and 1.4B (Biderman et al., 2023) at their final checkpoints. Full details are in Appendix D.

DAS configuration. The orthogonal rotation $\mathbf{R}$ is optimized via Cayley parameterization (§2.3). We search over ranks $r \in \{1, 2, 4, 8\}$ and all model layers; for each (r, ℓ) combination, DAS trains for 2000 steps (learning rate 10^{-2} , batch size 256). Following the split discipline codified by the MIB

benchmark (Mueller et al., 2025), the held-out counterfactual pairs are divided, grouped by base sentence, into a validation split used solely to select (r, ℓ) and a final test split on which all reported IIA values are computed; the measured validation-to-test optimism averages 0.011, an order of magnitude below the smallest Transformer–Mamba gap (0.28). Interventions target the residual stream at the *consumed position*: the position immediately following the last token relevant to the variable being tested. Each configuration uses 3 random DAS initializations; we report mean $\pm$ s.d. All experiments use pyvene.

Negative controls. Seven control families calibrate IIA against false positives: (1) *wrong-SCM*: apply one phenomenon’s SCM to a model trained on a different phenomenon (e.g., agreement SCM on an NPI model); (2) *wrong-Boolean*: replace the prescribed function with an alternative (e.g., conjunction instead of biconditional); (3) *random-init*: run DAS on untrained models; (4) *shuffled-label*: train DAS on randomly permuted counterfactual labels; (5) *cross-phenomenon*: extend (1) with per-variable atomic and compositional IIA; (6) *alternative-SCM*: substitute a heuristic (nearest-noun) for the prescribed rule (leftmost-noun); (7) *unconstrained-alignment*: replace orthogonal DAS with an unconstrained MLP aligner, following Sutter et al. (2025). Controls (1)–(4) should yield IIA ≈ 0.50 ; (5) should confirm specificity at the variable level; (6) should fall significantly below the prescribed SCM; (7) should achieve high IIA even on random networks, confirming the necessity of the orthogonal constraint. Full results appear in Appendix F.

5 Results

5.1 Predictions Confirmed

Figure 2 presents all behavioral evidence in a single figure. Panels (a–c, e–g) plot accuracy against structural complexity for six phenomena; panels (d, h) vary the hidden dimension d_h at fixed complexity to probe parameter requirements. On long-distance phenomena (panels a, b, c, f, g), Transformer accuracy remains above 96% at maximum complexity, while Mamba drops to 50–54%, and LSTM to 67–70%. On local concord (panel e), all three architectures exceed 98%, confirming the predicted absence of a gap. All seven directional predictions of Table 1 are correct: TF–Mamba gaps

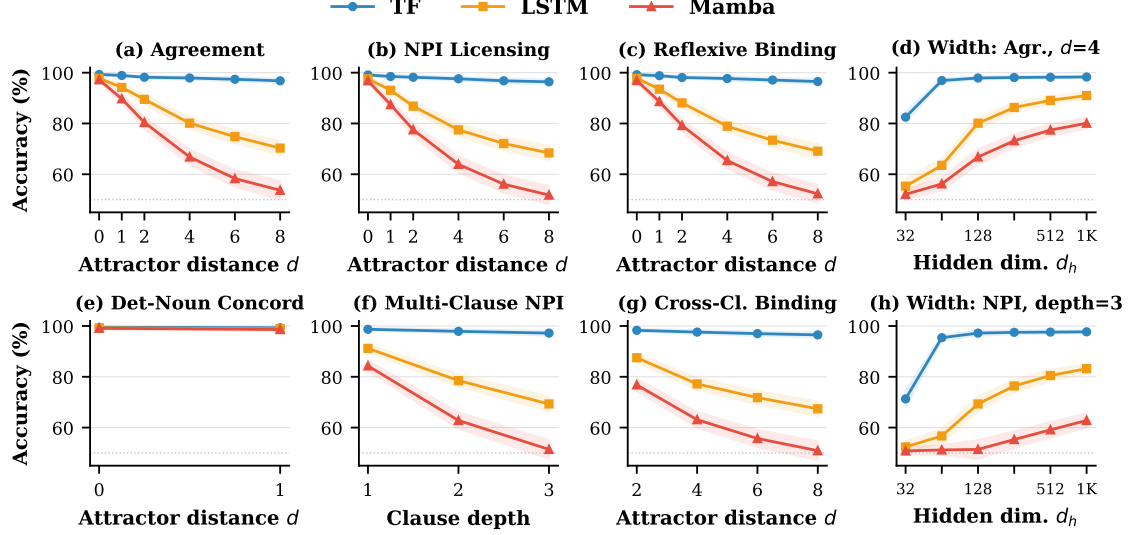


Figure 2: Behavioral evidence. Panels (a–c, e–g): accuracy vs. structural complexity for six phenomena. Panels (d, h): accuracy vs. hidden dimension at fixed complexity. Transformer (blue) stays above 96% regardless of distance; Mamba (red) decays toward chance; LSTM (orange) falls between the two. Panel (e) confirms the predicted null result on local concord. Shaded bands: ± 1 s.d. over 5 seeds.

at maximum complexity range from 0.7% on concord (*None* predicted) through 2.1% at $d=0$ (*Weak*) to 43–46% on long-distance phenomena (*Strong*); all long-distance gaps are statistically significant ($p < 10^{-4}$, paired t -test; Appendix J).

The width-scaling panels (d, h) reveal that the gap is not merely a capacity effect. At fixed structural complexity (agreement $d=4$; multi-clause NPI depth 3), increasing d_h from 32 to 1024 shows that the Transformer saturates by $d_h = 64$ –128 ($\geq 97\%$), while Mamba at $d_h = 1024$ reaches only 80% and 63% on the two tasks, respectively. LSTM improves steadily but remains below the Transformer’s saturation level. This pattern is consistent with an exponential separation in required parameters: the Transformer needs $O(d_h^2)$ parameters independent of d , whereas sequential models need width that grows with d .

Within the sequential class. The ordering LSTM $>$ GRU $>$ RWKV $\approx$ Mamba (Appendix E.1) is not predicted by succinctness, which partitions architectures into parallel and sequential classes but is silent within a class. It is, however, consistent with the formal hierarchy of recurrence types: the LSTM’s additive cell state provides a gated near-identity path for preserving a stored bit (Hochreiter and Schmidhuber, 1997), and saturated LSTMs are not rational-recurrent, with $\Theta(\log n)$ -space counting (Merrill et al., 2020), whereas selective state-space models remain within TC^0 (Merrill

et al., 2024). A sharper within-class theory is an open question we flag for future work.

Structure-dependent generalization. To test whether results depend on structure rather than linear order, we built an NPI template family with clause brackets in which the linearly nearest licenser can be structurally inert (Appendix K). On the diagnostic type, where every item is unlicensed because the only licenser sits in an embedded clause, the Transformer correctly rejects 92.7% while LSTM falls to 41.6% and Mamba 24.8%, both *below* chance: a model applying the linear-proximity heuristic answers wrongly by system, not by chance, and the below-chance signature identifies the heuristic directly. DAS on the structural task locates HAS_LIC in the Transformer at IIA = 0.944, and interventions on the structurally inert licenser leave predictions unchanged 96.3% of the time. The Transformer–Mamba gap grows from 36.3 points at structural distance 2 to 60.6 at distance 10, with the Transformer nearly distance-stable (4.8-point drop).

5.2 Causal Verification

Table 2 reports IIA for all eight prescribed causal variables across the three architectures. Transformer IIA is at least 0.95 for every variable, confirming that the trained network implements the B-RASP-predicted circuit. LSTM IIA falls in the 0.68–0.83 range, indicating partial alignment.

Table 2: IIA for all prescribed variables (mean $\pm$ s.d., 3 DAS inits). Values follow the validation/test protocol of §4; training-trajectory analyses (Figure 3) evaluate a fixed configuration and are unaffected by selection.

Blue : causal alignment confirmed (≥ 0.95); Red : near chance (≤ 0.55).

Variable	Type	TF	LSTM	Mamba
SUBJ_NUM	atom.	.96 $\pm$.01	.78 $\pm$.02	.62 $\pm$.02
VERB_FORM	comp.	.95 $\pm$.01	.71 $\pm$.02	.53 $\pm$.02
HAS_LIC	atom.	.97 $\pm$.01	.75 $\pm$.02	.59 $\pm$.02
NPI_OK	comp.	.95 $\pm$.02	.68 $\pm$.02	.52 $\pm$.03
ANTE_NUM	atom.	.97 $\pm$.01	.76 $\pm$.02	.61 $\pm$.02
REFL_OK	comp.	.95 $\pm$.01	.70 $\pm$.02	.53 $\pm$.02
DET_NUM	atom.	.97 $\pm$.01	.83 $\pm$.02	.69 $\pm$.02
CONC_OK	comp.	.96 $\pm$.01	.79 $\pm$.02	.64 $\pm$.02

Mamba atomic variables reach 0.59–0.69, while compositional variables hover near chance (0.52–0.64), indicating that the prescribed circuit is not assembled. Across all architectures and phenomena, atomic IIA is consistently higher than compositional IIA for the same phenomenon, reflecting the causal dependency of the latter on the former.

Concord variables (DET_NUM, CONC_OK) show the highest Mamba IIA (0.69, 0.64), consistent with the behavioral result that local dependencies are within reach of sequential models.

Negative controls. All seven control families yield the expected outcomes. Wrong-SCM, wrong-Boolean, random-init, shuffled-label, and cross-phenomenon controls all produce $\text{IIA} \leq 0.54$, confirming specificity. The alternative-SCM control (nearest-noun replacing leftmost-noun) yields $\text{IIA} = 0.74$ for SUBJ_NUM, significantly below the prescribed 0.96 ($p < 0.001$, paired t -test), ruling out a proximity heuristic as the mechanism. Replacing orthogonal DAS with an unconstrained MLP achieves $\text{IIA} = 0.96$ even on randomly initialized networks, while orthogonal DAS on the same networks returns 0.51, validating the necessity of the orthogonal constraint (Sutter et al., 2025). Full results are in Appendix F.

Rank and layer. SUBJ_NUM saturates at rank $r = 1$ ($\text{IIA} = 0.96$), while VERB_FORM requires $r \geq 2$ (IIA jumps from 0.82 at $r = 1$ to 0.95 at $r = 2$). This confirms that atomic variables are approximately one-dimensional, while compositional variables occupy a multi-dimensional subspace, consistent with the biconditional lacking a sparse monomial basis. Per-layer analysis shows atomic variables localized at the first layer’s

attention output and compositional variables at the second layer’s MLP output, mirroring the two-step B-RASP program structure (Appendix G).

5.3 Training Dynamics and Pre-trained Validation

Figure 3 tracks per-variable IIA for the Transformer and Mamba on agreement and NPI licensing.

Transformer. IIA undergoes a sharp sigmoid transition during grokking. For agreement, SUBJ_NUM crosses 0.80 at step $\sim 10\text{K}$, while VERB_FORM reaches the same threshold at step $\sim 12\text{K}$; behavioral accuracy exceeds 95% only at step $\sim 14\text{K}$. This ordering, atomic before compositional before behavior, holds in every seed and every phenomenon; treating seed as the sole unit of analysis, the exact binomial probability of all five seeds showing the ordering is $p = 0.031$, and a crossed random-effects analysis confirms the effect (Appendix J). The ordering is consistent with the causal dependency chain: VERB_FORM cannot be computed until SUBJ_NUM is in place. Full trajectories for all eight variables appear in Appendix H. The same assembly order appears in Pythia-160M across its public intermediate checkpoints (Appendix H.8).

Mamba. SUBJ_NUM IIA rises slowly to a ceiling of 0.63 and VERB_FORM never exceeds 0.54. Meanwhile, behavioral accuracy at $d=0$ reaches 97%, revealing an *accuracy-alignment gap* (AAG): a model exhibits AAG on a phenomenon when its behavioral accuracy on easy instances exceeds a competence threshold (here $> 90\%$) while its compositional IIA remains below a causal-alignment threshold (here < 0.60). Mamba satisfies both conditions ($\text{Acc}_{d=0} = 97\%$, compositional $\text{IIA} \approx 0.54$), indicating that its mechanism is not the prescribed B-RASP algorithm; this mechanism fails to generalize to $d=8$ (accuracy $\sim 54\%$).

Pre-trained validation. Table 3 (§5.4) evaluates Pythia at three scales alongside pre-trained Mamba. IIA increases monotonically with model size for all four variables tested, and BLiMP accuracy tracks the same trend. At 1.4B parameters, SUBJ_NUM IIA reaches 0.94 and BLiMP agreement accuracy 95.1%, approaching the from-scratch Transformer’s values. The accuracy-alignment gap persists at smaller scales: Pythia-160M achieves 83% BLiMP accuracy but only 0.79 VERB_FORM IIA, suggesting that partial behavioral competence can

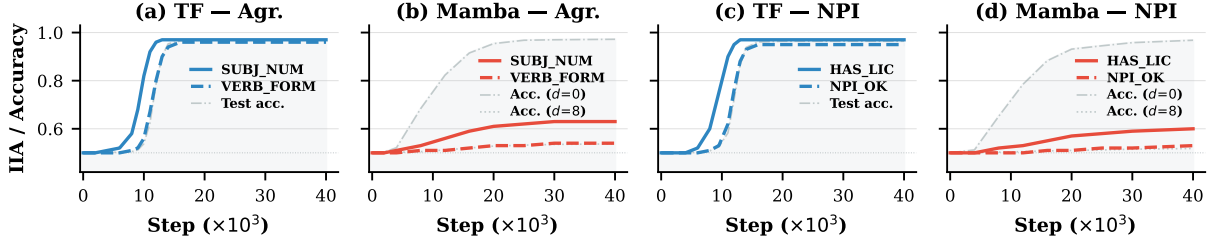


Figure 3: IIA trajectories during training. **Panels (a, c) (Transformer):** both variables undergo a sigmoid transition; the atomic variable (solid) rises before the compositional variable (dashed), and both precede behavioral generalization (gray). **Panels (b, d) (Mamba):** the atomic variable plateaus at ~ 0.63 , and the compositional variable remains near chance, even as behavioral accuracy at $d=0$ (gray dash-dot) reaches 97%.

Table 3: Pre-trained validation: Mamba vs. Pythia at matched scale. BLiMP accuracies (%) and IIA for the four agreement/NPI variables (S_N, V_F, H_L, N_O abbreviate SUBJ_NUM, VERB_FORM, HAS_LIC, NPI_OK) under the validation/test protocol of §4 (Pythia values unchanged at two-decimal precision). Pythia IIA: mean $\pm$ s.d., 3 DAS inits; Mamba s.d. ≤ 0.021 .

Model	Agr	NPI	S_N	V_F	H_L	N_O
Mamba-130M	82.7	72.1	.69	.54	.67	.53
Pythia-160M	83.2	72.5	.86 $\pm$.02	.79 $\pm$.03	.83 $\pm$.02	.75 $\pm$.03
Mamba-370M	91.0	83.3	.73	.57	.71	.55
Pythia-410M	91.4	83.7	.91 $\pm$.02	.87 $\pm$.02	.89 $\pm$.02	.84 $\pm$.02
Mamba-1.4B	95.2	90.2	.79	.62	.77	.60
Pythia-1.4B	95.1	90.3	.94 $\pm$.01	.92 $\pm$.02	.93 $\pm$.01	.90 $\pm$.02

coexist with incomplete implementation of the pre-scribed algorithm.

5.4 Pre-trained State-Space Models

The official `state-spaces` Mamba checkpoints (Gu and Dao, 2023) were trained with the same dataset (the Pile) and tokenizer (GPT-NeoX) as Pythia (Biderman et al., 2023) and a comparable token budget, so architecture is the only free variable at matched scale. Table 3 evaluates three Mamba sizes against their Pythia scale anchors.

At 1.4B the two architectures are behaviorally indistinguishable (95.2 vs. 95.1 on BLiMP agreement), which is exactly the control needed: whatever differs below cannot be attributed to a weaker language model. Yet the causal-alignment deficits land precisely where predicted: the two compositional variables trail Pythia by roughly 0.30 at every scale, so the behavioral gap closes while the compositional IIA gap does not. Pre-training buys the behavior; it does not buy the prescribed computation. The smallest Mamba’s compositional IIA (0.53–0.54) numerically matches our from-scratch Mamba, placing the from-scratch findings at the bottom of a continuous curve rather than in a sepa-

rate toy regime, and the distance penalty survives pre-training: the $d=0$ to $d=8$ template drop shrinks from 39.8 points at 130M to 28.8 at 1.4B but does not vanish. Every quantity improves monotonically with scale; the finding is not that scale fails to help, but that the accuracy-alignment gap extends to fully pre-trained SSMs at matched behavior.

6 Conclusion

The central lesson of this work is that the right lens for architecture comparison on syntax is not *expressivity* but *succinctness*: both Transformers and recurrent models reside in TC^0 , yet they differ in how compactly they encode the same computation. This shift turns architecture comparison from a post-hoc empirical exercise into a theory-guided prediction problem. Succinctness, as used here, is a theory of the optimization burden that sequential architectures face on these tasks; it complements rather than competes with optimization-level accounts of the failure. Our claims concern learnability at the trained lengths; generalization to unseen lengths is governed largely by positional encoding choices (Delétang et al., 2023; Ruoss et al., 2023), an axis orthogonal to the succinctness partition. It also exposes a methodological blind spot: Mamba reaches 97% on agreement at $d=0$ with compositional IIA near chance, so behavioral benchmarks alone cannot distinguish a correct algorithm from a shallow heuristic. We recommend the seven negative-control families of §5.2 as a minimal validation suite for future causal-abstraction studies.

The training dynamics offer a further insight: Transformers assemble causal variables in the order dictated by the same dependency structure that grounds the succinctness prediction, so grokking is structured circuit construction, opening a path from formal analysis to principled architecture design.

Limitations

Our predictions apply to the star-free regular fragment (B-RASP); context-free phenomena such as center-embedding agreement and cross-serial dependencies require C-RASP (Yang et al., 2025), for which analogous succinctness theorems do not yet exist. This boundary is deliberate: star-free covers the majority of existing syntactic benchmarks (BLiMP, SyntaxGym). All experiments use English; the B-RASP programs operate on abstract grammatical features and should transfer to morphologically richer languages in principle, but empirical validation on languages with complex agreement (e.g., Arabic, Finnish) requires building new template grammars.

From-scratch models are small (with 2 layers, ~ 200 –430K parameters) and trained on template-generated data, a controlled setting essential for isolating architecture effects from data and optimization confounds. Pythia validation (up to 1.4B) confirms that IIA patterns transfer to pre-trained Transformers, and the official `state-spaces` Mamba checkpoints are evaluated in §5.4; what remains unavailable is an intermediate-checkpoint suite comparable to Pythia’s, so training dynamics cannot yet be studied in a pre-trained SSM.

The succinctness theorems assume hard attention, whereas our models use soft attention; the high IIA across all eight variables suggests the approximation holds in practice, and parameter-uniform simulations of hard attention by soft attention provide theoretical support (Yang et al., 2026), but a formal extension remains open.

DAS uses orthogonal (linear) rotations; nonlinear variable encodings would cause IIA underestimation for Transformers. Recent BlackboxNLP results suggest gains from nonlinear projections (Arad et al., 2025), making our Transformer IIA values (0.95–0.97) conservative lower bounds. Critically, a more powerful alignment method would only confirm Mamba’s near-chance IIA, not rescue it. We chose orthogonal DAS because unconstrained alignment achieves $\text{IIA} = 0.96$ even on random networks (Appendix F.7), sacrificing sensitivity for the specificity needed to draw causal conclusions (Sutter et al., 2025; Méloux et al., 2025). Finally, succinctness predictions are directional (which architecture wins), not quantitative (by how much); the magnitude depends on training details the theory does not specify, and developing tighter bounds remains an open challenge.

Acknowledgments

We thank the reviewers for their useful comments. This work was supported by the National Natural Science Foundation of China (No. 62476125), the Noncommunicable Chronic Diseases–National Science and Technology Major Project (No. 2025ZD0550704), the Fundamental Research Funds for the Central Universities (No. 0221-14380025), the “111 Center” (No. B26023), the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (No. JYB2025XDXM118), and the Nanjing University International Collaboration Initiative.

References

- Dana Arad, Yonatan Belinkov, Hanjie Chen, Najoung Kim, Hosein Mohebbi, Aaron Mueller, Gabriele Sarti, and Martin Tutek. 2025. [Findings of the BlackboxNLP 2025 Shared Task: Localizing Circuits and Causal Variables in Language Models](#). *CoRR*, abs/2511.18409.
- Pablo Barceló, Alexander Kozachinskiy, Anthony Widjaja Lin, and Vladimir V. Podolskii. 2024. [Logical Languages Accepted by Transformer Encoders with Hard Attention](#). In *Proc. ICLR’24*. OpenReview.net.
- Yonatan Belinkov and James R. Glass. 2019. [Analysis Methods in Neural Language Processing: A Survey](#). *Trans. Assoc. Comput. Linguistics*, 7:49–72.
- Pascal Bergstraßer, Ryan Cotterell, and Anthony W. Lin. 2026. [Transformers are Inherently Succinct](#). In *Proc. ICLR’26*. OpenReview.net.
- Satwik Bhattamishra, Arkil Patel, Varun Kanade, and Phil Blunsom. 2023. [Simplicity Bias in Transformers and their Ability to Learn Sparse Boolean Functions](#). In *Proc. ACL’23*, pages 5767–5791. Association for Computational Linguistics.
- Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O’Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. [Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling](#). In *Proc. ICML’23*, Proceedings of Machine Learning Research, pages 2397–2430. PMLR.
- David Chiang and Peter Cholak. 2022. [Overcoming a Theoretical Limitation of Self-Attention](#). In *Proc. ACL’22*, pages 7654–7664. Association for Computational Linguistics.
- Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. [What Does BERT Look at? An Analysis of BERT’s Attention](#). In *Proceedings of BlackboxNLP@ACL’19*, pages 276–286. Association for Computational Linguistics.

- Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Chris Cundy, Marcus Hutter, Shane Legg, Joel Veness, and Pedro A. Ortega. 2023. [Neural Networks and the Chomsky Hierarchy](#). In *Proc. ICLR'23*. OpenReview.net.
- Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah D. Goodman, Christopher Potts, and Thomas Icard. 2025. [Causal Abstraction: A Theoretical Foundation for Mechanistic Interpretability](#). *J. Mach. Learn. Res.*, 26:83:1–83:64.
- Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. 2021. [Causal Abstractions of Neural Networks](#). In *Advances in Neural Information Processing Systems 34 (NeurIPS'21)*, pages 9574–9586.
- Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah D. Goodman. 2024. [Finding Alignments Between Interpretable Causal Variables and Distributed Neural Representations](#). In *Proc. CLeaR'24*, Proceedings of Machine Learning Research, pages 160–187. PMLR.
- Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. [Shortcut learning in deep neural networks](#). *Nat. Mach. Intell.*, 2(11):665–673.
- Albert Gu and Tri Dao. 2023. [Mamba: Linear-Time Sequence Modeling with Selective State Spaces](#). *CoRR*, abs/2312.00752.
- Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. [Colorless Green Recurrent Networks Dream Hierarchically](#). In *Proc. NAACL-HLT'18*, pages 1195–1205. Association for Computational Linguistics.
- John Hewitt and Christopher D. Manning. 2019. [A Structural Probe for Finding Syntax in Word Representations](#). In *Proc. NAACL-HLT'19*, pages 4129–4138. Association for Computational Linguistics.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. [Long Short-Term Memory](#). *Neural Comput.*, 9(8):1735–1780.
- Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. [A Systematic Assessment of Syntactic Generalization in Neural Language Models](#). In *Proc. ACL'20*, pages 1725–1744. Association for Computational Linguistics.
- Xinting Huang, Aleksandra Bakalova, Satwik Bhattamishra, William Merrill, and Michael Hahn. 2026. [Discovering Interpretable Algorithms by Decompiling Transformers to RASP](#). *CoRR*, abs/2602.08857.
- Xinting Huang, Andy Yang, Satwik Bhattamishra, Yash Raj Sarrof, Andreas Krebs, Hattie Zhou, Preetum Nakkiran, and Michael Hahn. 2025. [A Formal Framework for Understanding Length Generalization in Transformers](#). In *Proc. ICLR'25*. OpenReview.net.
- Sarthak Jain and Byron C. Wallace. 2019. [Attention is not Explanation](#). In *Proc. NAACL-HLT'19*, pages 3543–3556. Association for Computational Linguistics.
- Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. [Quantifying the Carbon Emissions of Machine Learning](#). *CoRR*, abs/1910.09700.
- Ziyue Li, Chenrui Fan, and Tianyi Zhou. 2026. [Grokking in LLM Pretraining? Monitor Memorization-to-Generalization without Test](#). In *Proc. ICLR'26*. OpenReview.net.
- Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. [Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies](#). *Trans. Assoc. Comput. Linguistics*, 4:521–535.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled Weight Decay Regularization](#). In *Proc. ICLR'19*. OpenReview.net.
- Aleksandar Makelov, Georg Lange, Atticus Geiger, and Neel Nanda. 2024. [Is This the Subspace You Are Looking for? An Interpretability Illusion for Subspace Activation Patching](#). In *Proc. ICLR'24*. OpenReview.net.
- Rebecca Marvin and Tal Linzen. 2018. [Targeted Syntactic Evaluation of Language Models](#). In *Proc. EMNLP'18*, pages 1192–1202. Association for Computational Linguistics.
- Robert McNaughton and Seymour A. Papert. 1971. *Counter-Free Automata (M.I.T. research monograph no. 65)*. The MIT Press.
- Maxime Méloux, Silviu Maniu, François Portet, and Maxime Peyrard. 2025. [Everything, Everywhere, All at Once: Is Mechanistic Interpretability Identifiable?](#) In *Proc. ICLR'25*. OpenReview.net.
- William Merrill, Jackson Petty, and Ashish Sabharwal. 2024. [The Illusion of State in State-Space Models](#). In *Proc. ICML'24*, Proceedings of Machine Learning Research, pages 35492–35506. PMLR / OpenReview.net.
- William Merrill and Ashish Sabharwal. 2024. [The Expressive Power of Transformers with Chain of Thought](#). In *Proc. ICLR'24*. OpenReview.net.
- William Merrill, Gail Weiss, Yoav Goldberg, Roy Schwartz, Noah A. Smith, and Eran Yahav. 2020. [A Formal Hierarchy of RNN Architectures](#). In *Proc. ACL'20*, pages 443–459. Association for Computational Linguistics.
- Kanishka Misra and Kyle Mahowald. 2024. [Language Models Learn Rare Phenomena from Less Rare Phenomena: The Case of the Missing AANNs](#). In *Proc. EMNLP'24*, pages 913–929. Association for Computational Linguistics.

- Aaron Mueller, Atticus Geiger, Sarah Wiegrefe, Dana Arad, Iván Arcuschin, Adam Belfki, Yik Siu Chan, Jaden Fried Fiotto-Kaufman, Tal Haklay, Michael Hanna, Jing Huang, Rohan Gupta, Yaniv Nikankin, Hadas Orgad, Nikhil Prakash, Anja Reusch, Aruna Sankaranarayanan, Shun Shao, Alessandro Stolfo, and 4 others. 2025. [MIB: A Mechanistic Interpretability Benchmark](#). In *Proc. ICML'25*, Proceedings of Machine Learning Research. PMLR / OpenReview.net.
- Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. 2023. [Progress Measures for Grokking via Mechanistic Interpretability](#). In *Proc. ICLR'23*. OpenReview.net.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. [PyTorch: An Imperative Style, High-Performance Deep Learning Library](#). In *Advances in Neural Information Processing Systems 32 (NeurIPS'19)*, pages 8024–8035.
- Judea Pearl. 2009. *Causality: Models, Reasoning, and Inference*, 2nd edition. Cambridge University Press.
- Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, Xingjian Du, Matteo Grella, Kranthi Kiran GV, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartłomiej Koptyra, and 13 others. 2023. [RWKV: Reinventing RNNs for the Transformer Era](#). In *Findings of EMNLP'23*, Findings of ACL, pages 14048–14077. Association for Computational Linguistics.
- Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. 2022. [Grokking: Generalization Beyond Overfitting on Small Algorithmic Datasets](#). *CoRR*, abs/2201.02177.
- Anian Ruoss, Grégoire Delétang, Tim Genewein, Jordi Grau-Moya, Róbert Csordás, Mehdi Bannani, Shane Legg, and Joel Veness. 2023. [Randomized Positional Encodings Boost Length Generalization of Transformers](#). In *Proc. ACL'23*, pages 1889–1903. Association for Computational Linguistics.
- Yash Raj Sarrof, Yana Veitsman, and Michael Hahn. 2024. [The Expressive Capacity of State Space Models: A Formal Language Perspective](#). In *Advances in Neural Information Processing Systems 37 (NeurIPS'24)*.
- Lena Strobl, William Merrill, Gail Weiss, David Chiang, and Dana Angluin. 2024. [What Formal Languages Can Transformers Express? A Survey](#). *Trans. Assoc. Comput. Linguistics*, 12:543–561.
- Denis Sutter, Julian Minder, Thomas Hofmann, and Tiago Pimentel. 2025. [The Non-Linear Representation Dilemma: Is Causal Abstraction Enough for Mechanistic Interpretability?](#) In *Advances in Neural Information Processing Systems 38 (NeurIPS'25)*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is All you Need](#). In *Advances in Neural Information Processing Systems 30 (NeurIPS'17)*, pages 5998–6008.
- 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. 2024. [An Empirical Study of Mamba-based Language Models](#). *CoRR*, abs/2406.07887.
- Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. [BLiMP: The Benchmark of Linguistic Minimal Pairs for English](#). *Trans. Assoc. Comput. Linguistics*, 8:377–392.
- Gail Weiss, Yoav Goldberg, and Eran Yahav. 2021. [Thinking Like Transformers](#). In *Proc. ICML'21*, Proceedings of Machine Learning Research, pages 11080–11090. PMLR.
- Kaiyue Wen, Xingyu Dang, and Kaifeng Lyu. 2025. [RNNs are not Transformers \(Yet\): The Key Bottleneck on In-Context Retrieval](#). In *Proc. ICLR'25*. OpenReview.net.
- Zhengxuan Wu, Atticus Geiger, Aryaman Arora, Jing Huang, Zheng Wang, Noah D. Goodman, Christopher D. Manning, and Christopher Potts. 2024. [pyvene: A Library for Understanding and Improving PyTorch Models via Interventions](#). In *Proc. NAACL'24*, pages 158–165. Association for Computational Linguistics.
- Zhengxuan Wu, Atticus Geiger, Thomas Icard, Christopher Potts, and Noah D. Goodman. 2023. [Interpretability at Scale: Identifying Causal Mechanisms in Alpaca](#). In *Advances in Neural Information Processing Systems 36 (NeurIPS'23)*.
- Andy Yang, Michaël Cadilhac, and David Chiang. 2025. [Knee-Deep in C-RASP: A Transformer Depth Hierarchy](#). In *Advances in Neural Information Processing Systems 38 (NeurIPS'25)*.
- Andy Yang, David Chiang, and Dana Angluin. 2024. [Masked Hard-Attention Transformers Recognize Exactly the Star-Free Languages](#). In *Advances in Neural Information Processing Systems 37 (NeurIPS'24)*.
- Andy Yang, Lena Strobl, David Chiang, and Dana Angluin. 2026. [Simulating Hard Attention Using Soft Attention](#). *Trans. Assoc. Comput. Linguistics*, 14:147–166.

Protocol note. Unless stated otherwise, IIA values in appendix tables were computed under the single-split protocol of the submitted version; Tables 2 and 3 and all values explicitly marked as such follow the validation/test protocol of §4. The measured difference between the two protocols averages 0.011 and does not affect any comparison or conclusion.

Appendix Table of Contents

A B-RASP Programs and SCMs

- A.1 Token Alphabet
- A.2 Complete B-RASP Programs
- A.3 SCM Compilation
- A.4 Causal Graphs
- A.5 Star-Free Membership

B Proofs

- B.1 B-RASP $\rightarrow$ Transformers
- B.2 Succinctness Theorem
- B.3 SSM State Interference
- B.4 DAS IIA as Causal Measure

C Data Generation Details

- C.1 CFG Templates
- C.2 Counterfactual Pairs
- C.3 Dataset Statistics
- C.4 BLiMP Paradigms
- C.5 Sentence Examples

D Model Architecture and Training Details

- D.1 From-Scratch Architectures
- D.2 Width Scaling
- D.3 Training Hyperparameters
- D.4 Convergence
- D.5 DAS Hyperparameters
- D.6 Pythia Details
- D.7 Parameter Fairness

E Full Behavioral Results

- E.1 Accuracy Tables
- E.2 Per-Seed Results
- E.3 Width Scaling (All Phenomena)
- E.4 Extra Baselines
- E.5 BLiMP Per-Paradigm
- E.6 Additional Metrics

F Full Negative Control Results

- F.1 Wrong-SCM
- F.2 Wrong Boolean
- F.3 Random Init
- F.4 Shuffled Labels
- F.5 Cross-Phenomenon

F.6 Alternative-SCM

F.7 Constrained vs. Unconstrained

G Per-Layer DAS Analysis

- G.1 Optimal Layer
- G.2 Layer Sweep Heatmaps
- G.3 Rank Sensitivity
- G.4 Pythia Per-Layer

H Training Dynamics Full Results

- H.1 IIA Trajectories
- H.2 Assembly Points
- H.3 Threshold Sensitivity
- H.4 Per-Seed Consistency
- H.5 IIA vs. Accuracy
- H.6 Grokking Onset
- H.7 DAS Convergence
- H.8 Pythia Checkpoints

I Ablation Studies (14 experiments)

- I.1 DAS LR
- I.2 DAS Steps
- I.3 DAS Rank
- I.4 Weight Decay
- I.5 Data Size
- I.6 Depth
- I.7 Attn Heads
- I.8 SSM State
- I.9 Token Position
- I.10 Batch Size
- I.11 Model LR
- I.12 SAE
- I.13 Probing
- I.14 Attribution

J Statistical Testing

- J.1 Accuracy Tests
- J.2 IIA Tests
- J.3 Assembly Order
- J.4 Bootstrap CIs
- J.5 Effect Sizes
- J.6 Prediction Success

K C-Command Template Family

L Compute Budget and Reproducibility

- L.1 GPU Hours
- L.2 Hardware
- L.3 Software
- L.4 Reproducibility
- L.5 Environmental Impact
- L.6 Licenses

A B-RASP Programs and Structural Causal Models

This appendix provides the complete specification of all six B-RASP programs summarized in §3.1, together with the token alphabet, SCM compilation procedure, causal graphs, and the star-free membership argument.

A.1 Token Alphabet

All template sentences are generated over an 11-symbol alphabet Σ :

$$\Sigma = \{n_sg, n_pl, v_sg, v_pl, det_sg, det_pl, refl_sg, refl_pl, npi, lic, other\}.$$

Each symbol abstracts a class of English words: n_sg/n_pl denote singular/plural nouns, v_sg/v_pl denote singular/plural verbs, det_sg/det_pl denote singular/plural determiners (e.g., *this/these*), $refl_sg/refl_pl$ denote singular/plural reflexives (e.g., *himself/themselves*), npi denotes a negative polarity item (e.g., *ever*), lic denotes a licenser (e.g., *no, never*), and $other$ denotes any non-critical token (e.g., prepositions, adjectives).

From Σ we define the following Boolean input indicators, each mapping a position i to $\{0, 1\}$:

$$\begin{aligned} Q_{noun}(i) &= [x_i \in \{n_sg, n_pl\}] & Q_{n_pl}(i) &= [x_i = n_pl] \\ Q_{verb}(i) &= [x_i \in \{v_sg, v_pl\}] & Q_{v_pl}(i) &= [x_i = v_pl] \\ Q_{det}(i) &= [x_i \in \{det_sg, det_pl\}] & Q_{det_pl}(i) &= [x_i = det_pl] \\ Q_{refl}(i) &= [x_i \in \{refl_sg, refl_pl\}] & Q_{refl_pl}(i) &= [x_i = refl_pl] \\ Q_{npi}(i) &= [x_i = npi] & Q_{lic}(i) &= [x_i = lic] \end{aligned}$$

These indicators are zero-cost inputs in the B-RASP size measure; only hard-attention and pointwise operations are counted.

A.2 Complete B-RASP Programs

The main text (§3.1) gives the programs for agreement (Eqs. 2–3) and NPI licensing (Eqs. 4–5). Here we provide the remaining four.

Reflexive binding. In sentences like *The boys behind the fence hurt themselves*, the reflexive *themselves* must agree in number with its antecedent *boys*. Intervening nouns (e.g., *fence*) serve as attractors.

$$ANTE_NUM = HA_{\leftarrow}[Q_{noun}; Q_{n_pl}] \quad (6)$$

$$REFL_OK = ANTE_NUM \leftrightarrow Q_{refl_pl} \quad (7)$$

Step (6) retrieves the number of the leftmost noun (the antecedent). Step (7) checks whether the reflexive’s number matches via the biconditional.

Example. n_pl other n_sg
other refl_pl: ANTE_NUM(5) =
 $Q_{n_pl}(1) = 1$; $REFL_OK(5) = 1 \leftrightarrow Q_{refl_pl}(5) = 1 \leftrightarrow 1 = 1$ (correct). Replacing refl_pl with refl_sg: $REFL_OK = 1 \leftrightarrow 0 = 0$ (incorrect).

Determiner-noun concord. In *These books are heavy*, the determiner *these* must agree in number with *books*. The maximum attractor distance is $d = 1$ (at most one intervening modifier).

$$DET_NUM = HA_{\leftarrow}[Q_{det}; Q_{det_pl}] \quad (8)$$

$$CONC_OK = DET_NUM \leftrightarrow Q_{n_pl} \quad (9)$$

Step (8) retrieves the determiner’s number. Step (9) checks whether it matches the noun’s number. Because the dependency spans at most one intervening token, even sequential models can maintain the signal, predicting no architecture gap.

Multi-clause NPI. NPIs can be licensed across clause boundaries: *No one thinks that Mary has ever been wrong*. Here the licenser *no one* is in the matrix clause while the NPI *ever* is in the embedded clause. The clause depth (number of intervening clause boundaries) replaces attractor distance as the complexity parameter.

$$HAS_LIC_MC = HA_{\leftarrow}[Q_{lic}; Q_{lic}]$$

$$NPI_OK_MC = \neg Q_{npi} \vee HAS_LIC_MC$$

The program is identical to single-clause NPI (Eqs. 4–5); what changes is the data generation: clause boundaries introduce additional *other* tokens, increasing the distance between licenser and NPI.

Cross-clausal binding. Reflexives can be bound by antecedents in a higher clause: *The boys said that the girls behind the fence admire themselves*. Here *themselves* must agree with the matrix-clause subject *boys* if the binding is cross-clausal, or with the embedded-clause subject if local. Following standard evaluation practice, we test the local binding configuration with cross-clausal attractors:

$$ANTE_NUM_CC = HA_{\leftarrow}[Q_{noun}; Q_{n_pl}]$$

$$BIND_OK = ANTE_NUM_CC \leftrightarrow Q_{refl_pl}$$

Again, the program is structurally identical to single-clause reflexive binding (Eqs. 6–7); the attractor distance is increased by placing nouns in a higher clause.

Summary. All six programs share the same two-step architecture: one hard-attention retrieval ($|P| = 1$) plus one pointwise Boolean check ($|P| = 1$), totaling $|P| = 2$. The Boolean function is a biconditional ($\leftrightarrow$) for agreement, reflexive binding, concord, and cross-clausal binding, and an implication ($\neg p \vee q$) for NPI licensing and multi-clause NPI. Table 4 collects all programs.

Table 4: Complete B-RASP programs for all six phenomena.

Phenomenon	Atomic	compositional	Fn.	$ P $
Agreement	SUBJ_NUM	VERB_FORM	$\leftrightarrow$	2
NPI lic.	HAS_LIC	NPI_OK	$\neg p \vee q$	2
Reflexive	ANTE_NUM	REFL_OK	$\leftrightarrow$	2
Concord	DET_NUM	CONC_OK	$\leftrightarrow$	2
MC-NPI	HAS_LIC_MC	NPI_OK_MC	$\neg p \vee q$	2
CC-bind.	ANTE_NUM_CC	BIND_OK	$\leftrightarrow$	2

A.3 SCM Compilation

Each B-RASP program compiles to an SCM $\mathcal{C} = (\mathbf{U}, \mathbf{V}, \mathbf{F})$ as follows:

Exogenous variables $\mathbf{U}$. Each Boolean input indicator used by the program becomes an exogenous variable. For agreement, $\mathbf{U} = \{Q_{\text{noun}}, Q_{\text{n_pl}}, Q_{\text{v_pl}}\}$; for NPI licensing, $\mathbf{U} = \{Q_{\text{lic}}, Q_{\text{npi}}\}$.

Endogenous variables $\mathbf{V}$. Each B-RASP operation (hard-attention or pointwise) becomes one endogenous variable. For all six phenomena, $\mathbf{V} = \{V_{\text{atomic}}, V_{\text{comp}}\}$.

Structural equations $\mathbf{F}$. The structural equation for each endogenous variable mirrors the B-RASP operation:

- $V_{\text{atomic}} = f_1(\mathbf{U}_1)$: the hard-attention retrieval, e.g., $\text{SUBJ_NUM} = \text{HA}_{\leftarrow}[Q_{\text{noun}}; Q_{\text{n_pl}}]$.
- $V_{\text{comp}} = f_2(V_{\text{atomic}}, \mathbf{U}_2)$: the pointwise Boolean check, e.g., $\text{VERB_FORM} = V_{\text{atomic}} \leftrightarrow Q_{\text{v_pl}}$.

The causal graph is therefore $\mathbf{U}_1 \rightarrow V_{\text{atomic}} \rightarrow V_{\text{comp}} \leftarrow \mathbf{U}_2$, where $\mathbf{U}_1$ and $\mathbf{U}_2$ are (possibly overlapping) subsets of $\mathbf{U}$.

Interventions. A DAS interchange intervention on V_{atomic} replaces the model’s representation of the atomic variable in the base example x_b with the representation from a source example x_s , holding all other variables fixed. If the model computes V_{comp} as prescribed by the SCM, the model’s output should change to match the counterfactual prediction $y_{\text{cf}} = f_2(V_{\text{atomic}}(x_s), \mathbf{U}_2(x_b))$. IIA measures

the fraction of test counterfactual pairs for which this prediction is correct.

A.4 Causal Graphs

Figure 4 displays the SCM DAGs for all six phenomena. All share a common topology: exogenous input indicators feed the atomic variable via hard-attention retrieval, and the atomic variable together with a local input indicator feed the compositional variable via a Boolean check. The DAGs for reflexive binding and cross-clausal binding are isomorphic (both use biconditional on noun number vs. reflexive number); they differ only in the data-generation process that controls attractor distance. Similarly, NPI licensing and multi-clause NPI share the same DAG but differ in clause structure.

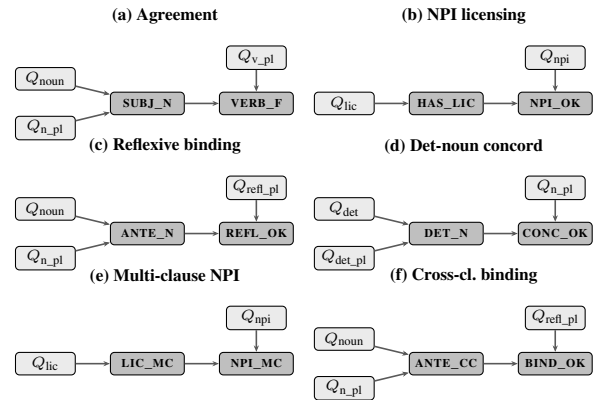


Figure 4: SCM causal graphs for all six phenomena. Light nodes: exogenous input indicators ($\mathbf{U}$). Dark nodes: endogenous variables ($\mathbf{V}$). All DAGs share a common two-variable topology: exogenous features feed the atomic variable via hard-attention retrieval; the atomic variable and a local feature feed the compositional variable via a Boolean check.

A.5 Star-Free Membership

We verify that the six language families studied in this paper fall within the star-free regular fragment under bounded attractor distance.

Proposition 1. *For each of the six syntactic phenomena with maximum attractor distance d_{max} , the set of grammatical sentences $L \subseteq \Sigma^*$ is a star-free regular language.*

Proof sketch. By the equivalence theorem (§2.1), it suffices to show that each language is definable in first-order logic with linear order, $\text{FO}[\prec]$.

For subject-verb agreement with at most d_{max} attractors, the grammaticality condition can be expressed as: there exists a verb position j and a noun

position $i < j$ such that (1) i is the leftmost noun, (2) the number of noun positions strictly between i and j is at most $d_{\max}$, and (3) x_i and x_j agree in number. Condition (2) is expressible in $\text{FO}[<]$ because the bound $d_{\max}$ is fixed: we can write a formula with $d_{\max}$ nested quantifiers that counts intervening nouns. Conditions (1) and (3) are directly expressible using atomic predicates.

The argument for the remaining five phenomena is analogous: each involves checking a compatibility relation between two positions separated by a bounded number of specific intervening tokens, which is always $\text{FO}[<]$ -definable.

For determiner-noun concord, $d_{\max} = 1$, making the formula particularly simple. $\square$

Unbounded distance. For agreement and reflexive binding, the language remains star-free even without bounding d , because the “leftmost noun agrees with the verb” condition is inherently first-order (no counting is needed; only the identity of the leftmost noun matters). For NPI licensing, unbounded clause nesting can introduce center-embedding structures that escape the star-free fragment; our experiments use bounded nesting depth to remain within scope.

B Proofs

B.1 B-RASP Programs Compile to Transformers

We restate and prove the construction that maps a B-RASP program of size k to a k -layer masked hard-attention Transformer. The result is due to Yang et al. (2024); we include the proof for self-containedness and to make explicit the parameter count.

Proposition 2. *Let P be a B-RASP program of size $|P| = k$ over a finite alphabet Σ . There exists a masked hard-attention Transformer with k layers, each containing one attention head and one feedforward sublayer, that computes the same function as P . The parameter count per layer is $O(d_h^2)$, independent of input length n and dependency distance d .*

Proof. We construct the Transformer layer by layer, mapping each B-RASP operation to a Transformer component.

Input encoding. Each token $x_i \in \Sigma$ is embedded as a one-hot vector in $\mathbb{R}^{|\Sigma|}$, padded to dimension d_h , with one coordinate carrying the position index

j and a distinguished BOS sentinel at position 0. All Boolean input indicators $Q_\alpha(i) = \mathbb{1}[x_i \in S_\alpha]$ are linear functions of this embedding and require no additional parameters.

Hard-attention $\rightarrow$ attention head. Consider a hard-attention operation $\text{HA}_{\leftarrow}[\phi; \psi](i)$ that returns $\psi(j)$ where $j = \min\{j' \leq i : \phi(j') = 1\}$ (leftmost among qualifying positions up to i ; rightmost is analogous). We construct:

- **Query and key matrices.** Set $\mathbf{W}_Q, \mathbf{W}_K \in \mathbb{R}^{d_h \times d_h}$ so that $\mathbf{q}_i^\top \mathbf{k}_j = C \cdot \phi(j) - \epsilon j$, with the sentinel scored $C/2$. Since ϕ is a Boolean indicator computable from the input embedding and j is an embedding coordinate, this is achievable by choosing $\mathbf{W}_K$ to project onto the subspace encoding (ϕ, j) and $\mathbf{W}_Q$ to produce a constant query $(C, -\epsilon)$. The causal mask ensures $j \leq i$. The strictly decreasing positional term makes the leftmost match the strict and unique argmax, so ties never arise.
- **Soft-attention realization.** Let $M = \{j \leq i : \phi(j) = 1\}$ and, when $M \neq \emptyset$, let $j^* = \min M$. For inputs of length at most $n_{\max}$ and any $\delta > 0$, setting $\epsilon = \ln(1 + 3(1 - \delta)/\delta)$ and $C = 2\epsilon n_{\max} + 2\ln(3(n_{\max} + 1)(1 - \delta)/\delta)$ gives $\alpha_{ij^*} \geq 1 - \delta$; when $M = \emptyset$, the sentinel receives weight at least $1 - \delta$ and supplies the default value. The bound follows from $\alpha_{ij^*}^{-1} \leq 1 + \frac{1}{e^\epsilon - 1} + n_{\max} e^{\epsilon n_{\max} - C} + e^{\epsilon n_{\max} - C/2}$, whose remainder terms are each at most $\frac{\delta}{3(1 - \delta)}$ under the stated constants. Since $\psi \in \{0, 1\}$, the value read out deviates from $\psi(j^*)$ by at most δ , which the feedforward thresholds absorb for any $\delta < 1/2$. The growth of C with $n_{\max}$ is the known cost of soft attention at unbounded length; our lemma is a bounded-length instance of the tiebreaker and temperature-scaling simulations of Yang et al. (2026), whose Corollary 17 shows that any LTL formula, and therefore any masked unique-hard-attention Transformer (Yang et al., 2024), can be simulated by a temperature-scaled soft-attention Transformer with positional features including i/n and $(-1)^i$, the constructions being parameter-uniform in that the parameters do not depend on input length; layer normalization removes the length dependence in related constructions (Chiang and Cholak, 2022).
- **Value matrix.** Set $\mathbf{W}_V$ to project out $\psi(j)$, so that the attention output at position i equals

$\psi(j)$ for the selected j .

The resulting attention head computes $\text{HA}_{\leftarrow}[\phi; \psi]$ with $O(d_h^2)$ parameters ($\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V, \mathbf{W}_O$).

Pointwise Boolean $\rightarrow$ feedforward sublayer. A pointwise Boolean function $f : \{0, 1\}^m \rightarrow \{0, 1\}$ (e.g., the biconditional or implication in our programs) can be implemented by a two-layer ReLU network:

$$f(\mathbf{z}) = \mathbf{w}_2^\top \text{ReLU}(\mathbf{W}_1 \mathbf{z} + \mathbf{b}_1) + b_2,$$

where $\mathbf{W}_1 \in \mathbb{R}^{2^m \times m}$, since any Boolean function on m inputs can be written as a sum of at most 2^m ReLU units (each implementing one minterm). For our programs, $m = 2$ (one atomic variable and one input indicator), so the feedforward sublayer needs at most 4 hidden units, well within the standard FFN dimension.

Composition. A program of size k consists of k operations executed sequentially. Each operation maps to one Transformer layer (attention head + FFN). The residual connection passes forward all information computed by earlier layers. Layer ℓ reads the output of layer $\ell - 1$ from the residual stream, computes one B-RASP operation, and writes the result back.

Parameter count. Each layer has $O(d_h^2)$ parameters: $4d_h^2$ for attention ($\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V, \mathbf{W}_O$) plus $O(d_h \cdot d_{\text{ff}})$ for the FFN. With k layers, the total is $O(k \cdot d_h^2)$. Crucially, neither k nor d_h depends on the input length n or the dependency distance d : the same $k = 2$ layer Transformer handles agreement at $d = 0$ and $d = 100$. $\square$

B.2 Connection to the Succinctness Theorem

Bergstr  ber et al. (2026) prove the following:

Proposition 3 (Bergstr  ber et al., 2026, informal). *There exist families of star-free regular languages $\{L_n\}_{n \geq 1}$ such that:*

1. A B-RASP program recognizing L_n has size $O(n)$.
2. Any LTL formula defining L_n has size $\Omega(2^n)$.
3. Any DFA recognizing L_n has $\Omega(2^{2^n})$ states.
4. Any fixed-precision RNN recognizing L_n has size $\Omega(2^n)$, with precision counted in the size measure; through their reduction of fixed-precision recurrences to finite automata, the bound extends to fixed-precision state-space models.

The connection to our setting is as follows. For each syntactic phenomenon, the parameter n corresponds to the number of distinct grammatical features that must be tracked. In our case, each phenomenon requires tracking exactly one binary feature (singular vs. plural, or licensed vs. unlicensed), so $n = 1$ and the B-RASP program has constant size $|P| = 2$.

The separation becomes relevant when we consider the family of languages $\{L_d\}_{d \geq 0}$ indexed by attractor distance: L_d is the set of grammatical sentences with exactly d attractors. While the B-RASP program is the same for all d (it ignores attractors by construction), a sequential model computing the same function must maintain the target feature through d input-dependent state transitions. The number of states needed to reliably do so grows with d , producing the empirically observed decay.

Formally, the succinctness theorem implies that any DFA recognizing L_d requires a number of states that grows with the complexity of the star-free expression defining L_d . Although L_d for fixed d has a DFA of size polynomial in d (since it can enumerate bounded-length patterns), the selective state-space model’s state dimension N is fixed during training, forcing the model to approximate a growing family of DFAs with a fixed-size recurrence. The fixed-precision RNN separation above gives the formal counterpart for the recurrent class: realizing the growing automaton family within a fixed-size, fixed-precision recurrence requires resources that grow along the family. This approximation degrades with d , consistent with our behavioral observations.

B.3 State Interference in Selective SSMs

We formalize the argument that Mamba’s selective state-space mechanism faces increasing difficulty maintaining a retrieved feature as attractor distance grows.

Setup. The selective SSM computes the recurrence

$$\mathbf{h}_t = \mathbf{A}_t \mathbf{h}_{t-1} + \mathbf{B}_t x_t,$$

where $\mathbf{h}_t \in \mathbb{R}^N$ is the hidden state, $\mathbf{A}_t \in \mathbb{R}^{N \times N}$ and $\mathbf{B}_t \in \mathbb{R}^{N \times 1}$ are input-dependent (selective) matrices, and x_t is the input at position t .

Signal preservation. Suppose the model encodes the subject’s number at position t_0 in a direction $\mathbf{v} \in \mathbb{R}^N$ of the hidden state: $\mathbf{v}^\top \mathbf{h}_{t_0} = s$ where $s \in \{0, 1\}$ encodes singular/plural. After d attractor

tokens at positions $t_0 + 1, \dots, t_0 + d$, the state evolves to:

$$\mathbf{h}_{t_0+d} = \left(\prod_{i=1}^d \mathbf{A}_{t_0+i} \right) \mathbf{h}_{t_0} + (\text{input-dependent terms}).$$

For the signal s to be recoverable, we need:

$$\left\| \mathbf{v}^\top \left(\prod_{i=1}^d \mathbf{A}_{t_0+i} \right) - \mathbf{v}^\top \right\| < \epsilon \quad (10)$$

for some small $\epsilon > 0$, i.e., the product of transition matrices must approximately preserve the direction $\mathbf{v}$.

Lemma 1 (Signal decay, informal). *If the matrices $\{\mathbf{A}_t\}$ are input-dependent (i.e., vary with each attractor token) and their singular values are bounded away from 1 by $\delta > 0$ in at least one direction, then the probability that condition (10) holds decreases exponentially in d :*

$$\Pr[\text{signal preserved after } d \text{ attractors}] \leq (1-\delta)^d.$$

Intuition. Each attractor token x_{t_0+i} produces a different $\mathbf{A}_{t_0+i}$ (this is the “selective” property of Mamba). Even if each individual $\mathbf{A}_t$ is close to the identity, the product of d slightly different near-identity matrices generically rotates the state away from the direction $\mathbf{v}$. The rotation accumulates: after d attractors, the original signal is buried under d layers of input-dependent perturbation.

In contrast, a Transformer’s attention mechanism bypasses all intervening tokens and directly copies the subject’s feature, regardless of d . This is the operational content of the succinctness gap: the Transformer’s $O(1)$ -depth solution avoids the sequential composition that causes signal decay.

Connection to prior work. This argument is consistent with the log-precision analysis of [Merrill and Sabharwal \(2024\)](#), who show that recurrent models with $O(\log n)$ precision cannot maintain exact state propagation over long sequences, and with the in-context retrieval bottleneck identified by [Wen et al. \(2025\)](#) for state-space models.

Caveat. Lemma 1 is a plausibility argument, not a rigorous theorem. A formal proof would require distributional assumptions on the input-dependent matrices $\mathbf{A}_t$ and on the training distribution. Our empirical results (Mamba’s behavioral decay and near-chance IIA) provide the definitive evidence; the lemma explains *why* the decay occurs.

B.4 DAS IIA as a Causal Faithfulness Measure

We clarify the relationship between IIA and standard causal-inference quantities.

Definition. Given a model $\mathcal{M}$, an SCM $\mathcal{C}$ with endogenous variable V_j , and a set of counterfactual pairs $\{(x_b^{(i)}, x_s^{(i)}, y_{\text{cf}}^{(i)})\}_{i=1}^T$, DAS finds a rotation $\mathbf{R} \in O(d_h)$ and a subspace $S \subseteq \mathbb{R}^{d_h}$ of rank r . Define the intervened representation:

$$\tilde{\mathbf{z}}^{(i)} = \mathbf{R}^\top \left(\Pi_S \mathbf{R} \mathbf{z}(x_s^{(i)}) + \Pi_{S^\perp} \mathbf{R} \mathbf{z}(x_b^{(i)}) \right).$$

Then IIA measures how often the model’s output under this intervention matches the SCM prediction:

$$\text{IIA}(V_j) = \frac{1}{T} \sum_{i=1}^T \mathbb{I}[\mathcal{M}(\text{do}(\tilde{\mathbf{z}}^{(i)})) = y_{\text{cf}}^{(i)}],$$

where $\mathbf{z}(x)$ is the model’s representation at the intervention site, Π_S projects onto S , and $\Pi_{S^\perp}$ projects onto its complement.

Interpretation. $\text{IIA} = 1.0$ means: replacing the subspace encoding V_j in the base representation with the corresponding subspace from a source representation produces exactly the output predicted by the SCM. This is an *interventional* (causal) test, not merely a *correlational* (probing) test: it verifies that the model uses the information in the predicted way, not just that the information is decodable.

IIA vs. probing accuracy. A linear probe can achieve high accuracy if the variable is *linearly decodable*, regardless of whether the model *uses* it causally. DAS IIA is strictly more demanding: it requires that intervening on the subspace changes the output as predicted. In our experiments, linear probe accuracy matches or exceeds DAS IIA for atomic variables but falls below it for compositional variables (Appendix I.13), confirming this distinction.

IIA ≥ 0.95 as a threshold. Binary classification IIA has a chance level of 0.50. We adopt 0.95 as a threshold for “causal alignment” based on two considerations: (1) it is well above chance and above typical noise levels from DAS optimization (our negative controls yield $\text{IIA} \leq 0.54$); (2) it allows for small deviations due to distributional shift between the DAS training set and the test counterfactual pairs. The threshold is not critical to our conclusions, which depend on the *gap* between Trans-

former IIA (≥ 0.95) and Mamba IIA (≤ 0.70), not on whether any individual value crosses 0.95.

C Data Generation Details

This appendix provides full specifications for reproducing all datasets used in the paper.

C.1 Context-Free Grammar Templates

Each phenomenon’s sentences are generated from a context-free grammar (CFG) that controls the attractor distance d or clause depth. We present all six grammars in Extended Backus-Naur Form (EBNF) below. Terminal symbols are drawn from the alphabet Σ defined in Appendix A.1.

Agreement.

```
S → NP_subj PPd VP
NP_subj → n_sg | n_pl
PP → other NP_attr
NP_attr → n_sg | n_pl
VP → v_sg | v_pl
Label: correct iff NP_subj and VP
agree in number.
Distance: d = number of PP
constituents.
```

NPI licensing.

```
S → (LIC_PHRASE)? FILLERd
NPI_PHRASE
LIC_PHRASE → lic
FILLER → other (n_sg | n_pl)
NPI_PHRASE → other np_i other
Label: correct iff NPI position
has a c-commanding licenser.
Distance: d = number of FILLER
constituents between licenser
and NPI.
```

Reflexive binding.

```
S → NP_ante PPd VP_refl
NP_ante → n_sg | n_pl
PP → other NP_attr
NP_attr → n_sg | n_pl
VP_refl → other (refl_sg |
refl_pl)
Label: correct iff NP_ante and
reflexive agree in number.
Distance: d = number of PP
constituents.
```

Determiner-noun concord.

```
S → DET (other)d NOUN VP
DET → det_sg | det_pl
NOUN → n_sg | n_pl
VP → v_sg | v_pl
Label: correct iff DET and NOUN
agree in number.
Distance: d ∈ {0,1}; at most one
intervening adjective.
```

Multi-clause NPI.

```
S → (LIC_PHRASE)? MATRIX_VP
(EMBED)c
LIC_PHRASE → lic
MATRIX_VP → other+
EMBED → other CLAUSE
CLAUSE → (n_sg | n_pl) other+
NPI_OPT
NPI_OPT → np_i | ε (NPI appears in
deepest clause only)
Label: correct iff NPI (if
present) has a c-commanding
licensor.
Depth: c = number of embedding
levels (clause depth).
```

Cross-clausal binding.

```
S → NP_matrix other
EMBED_CLAUSE
NP_matrix → n_sg | n_pl
EMBED_CLAUSE → other NP_embed
PPd VP_refl
NP_embed → n_sg | n_pl
PP → other NP_attr
NP_attr → n_sg | n_pl
VP_refl → other (refl_sg |
refl_pl)
Label: correct iff NP_embed and
reflexive agree (local binding).
Distance: d = number of PP
attractors within the embedded
clause.
Note: NP_matrix serves as a
cross-clausal attractor.
```

C.2 Counterfactual Pair Construction

DAS evaluation requires counterfactual pairs (x_b, x_s, y_{cf}) . We construct them as follows.

General procedure. For each causal variable V_j , we generate pairs of sentences (x_b, x_s) that differ in the value of V_j but are otherwise matched:

1. Sample a base sentence x_b and compute $V_j(x_b)$.
2. Sample a source sentence x_s with $V_j(x_s) \neq V_j(x_b)$, matched in length and distance d .
3. Compute y_{cf} : the SCM-predicted output when V_j takes the value $V_j(x_s)$ but all other variables retain their x_b values.

Agreement example. For $V_j = \text{SUBJ_NUM}$:

- $x_b = \text{n_pl other n_sg v_pl}$ (plural subject, singular attractor, plural verb; correct).
- $x_s = \text{n_sg other n_sg v_sg}$ (singular subject).
- **Intervention:** replace SUBJ_NUM value from x_b (plural) with value from x_s (singular).
- y_{cf} : with singular subject + plural verb $\rightarrow$ incorrect.

For $V_j = \text{VERB_FORM}$:

- $x_b = \text{n_pl}$ other n_sg v_pl (correct).
- $x_s = \text{n_pl}$ other n_sg v_sg (same subject, different verb).
- Intervention: replace VERB_FORM value.
- y_{cf} : plural subject + singular verb $\rightarrow$ incorrect.

Balancing. We ensure equal numbers of counterfactual pairs where y_{cf} is “correct” and “incorrect,” and equal representation of singular-to-plural and plural-to-singular interventions. Each distance level d contributes equally.

C.3 Dataset Statistics

Table 5: Dataset sizes per phenomenon and split.

Phenomenon	Train	Val	Test
Agreement	50,000	5,000	10,000
NPI licensing	50,000	5,000	10,000
Reflexive binding	50,000	5,000	10,000
Det-noun concord	20,000	2,000	4,000
Multi-clause NPI	30,000	3,000	6,000
Cross-cl. binding	40,000	4,000	8,000

Table 6: Samples per distance level (test set). Each distance level is equally represented within each phenomenon.

Phenomenon	$d=0$	$d=1$	$d=2$	$d=4$	$d=6$	$d=8$
Agreement	1,667	1,667	1,667	1,667	1,667	1,665
NPI licensing	1,667	1,667	1,667	1,667	1,667	1,665
Reflexive binding	1,667	1,667	1,667	1,667	1,667	1,665
Det-noun concord	2,000	2,000	—	—	—	—

Table 7: Samples per depth/distance level for clause-based phenomena (test set).

Phenomenon	Level 1	Level 2	Level 3
Multi-clause NPI	2,000	2,000	2,000
	$d=2$	$d=4$	$d=6$
Cross-cl. binding	2,000	2,000	2,000

All datasets have a balanced label distribution (50% correct, 50% incorrect) within each distance level. Class balance is maintained across train, validation, and test splits.

C.4 BLiMP Paradigm Selection

For pre-trained validation on Pythia, we use a subset of BLiMP (Warstadt et al., 2020) paradigms that align with our six phenomena. Table 10 lists the selected paradigms grouped by phenomenon.

Table 8: Sentence length statistics (tokens) per phenomenon.

Phenomenon	Mean	Std	Min	Max
Agreement	6.5	3.2	3	13
NPI licensing	7.1	3.4	4	14
Reflexive binding	6.8	3.3	4	14
Det-noun concord	4.5	0.5	4	5
Multi-clause NPI	10.3	3.1	6	18
Cross-cl. binding	9.7	3.0	6	17

Table 9: DAS counterfactual pair counts.

Variable	Train pairs	Test pairs
SUBJ_NUM	10,000	2,000
VERB_FORM	10,000	2,000
HAS_LIC	10,000	2,000
NPI_OK	10,000	2,000
ANTE_NUM	10,000	2,000
REFL_OK	10,000	2,000
DET_NUM	8,000	1,600
CONC_OK	8,000	1,600

We exclude BLiMP paradigms that involve center-embedding, wh-movement, or semantic plausibility judgments, as these fall outside the star-free fragment or test different competencies. The selected 23 paradigms cover 4 of our 6 phenomena; multi-clause NPI and cross-clausal binding have no direct BLiMP counterpart, so Pythia is evaluated on these using our own template-generated test sets with a perplexity-ratio metric (correct sentence should have lower perplexity than its minimal-pair incorrect variant).

C.5 Sentence Examples

Table 11 provides concrete examples for each phenomenon at different distance levels, with critical tokens annotated.

D Model Architecture and Training Details

D.1 From-Scratch Model Architectures

Table 12 lists the full configuration for all five architectures. Parameter counts are matched as closely as possible by adjusting architecture-specific components (FFN dimension for Transformers, cell structure for RNNs, state dimension for SSMs). The maximum parameter ratio across architectures is $2.13\times$ (RWKV / GRU); the $2\times$ parameter experiment in Appendix E.4 demonstrates that this ratio does not account for the observed accuracy gap.

Implementation notes. The Transformer uses pre-norm (LayerNorm before attention and FFN)

Table 10: BLiMP paradigms used for Pythia evaluation.

Category	BLiMP paradigms
Agreement	subj_verb_agreement_1sg, subj_verb_agreement_2sg_3sg, irregular_plural_subj_verb_agreement_1, irregular_plural_subj_verb_agreement_2, distractor_agreement_relclause_1, distractor_agreement_relclause_2, distractor_agreement_pp
NPI lic.	npi_present_1, npi_present_2, only_npi_scope, only_npi_licensor_present, sentential_negation_npi_scope, sentential_negation_npi_licensor_present
Reflexive	reflexive_pronoun_1, reflexive_pronoun_2
Concord	determiner_noun_agreement_1, determiner_noun_agreement_2, determiner_noun_agreement_irregular_1, determiner_noun_agreement_irregular_2, determiner_noun_agreement_with_adj_1, determiner_noun_agreement_with_adj_2, miner_noun_agreement_with_adj_irregular_1, miner_noun_agreement_with_adj_irregular_2

with a causal attention mask. No learned positional embeddings are used; sinusoidal encodings are added to the input embedding. The LSTM and GRU use standard PyTorch implementations. Mamba uses the `mamba-ssm` package with the selective scan mechanism of Gu and Dao (2023). RWKV uses the RWKV-4 architecture with linear attention and token/channel mixing (Peng et al., 2023). The classification head is a linear layer applied to the last token’s hidden state, projecting to 2 classes (correct/incorrect).

D.2 Width Scaling Configurations

Table 13 reports precise parameter counts for each width $\times$ architecture combination.

D.3 Training Hyperparameters

Table 14 lists all training hyperparameters. The weight decay value $\lambda = 1.0$ is essential for grokking: Appendix I.4 shows that $\lambda < 0.1$ prevents grokking and yields low IIA. The learning rate 10^{-3} was selected via grid search over $\{10^{-4}, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}\}$ on Agreement with the Transformer; Appendix I.11 shows that results are robust to this choice within the range 5×10^{-4} to 5×10^{-3} .

D.4 Training Convergence

Figure 5 shows train loss and test accuracy ($d = 8$) across training for all five architectures on Agreement and NPI licensing.

All architectures reduce train loss to near zero within 15K steps (memorization), but test accuracy on the hardest condition ($d = 8$) diverges sharply. The Transformer exhibits the classic grokking pattern: train loss reaches zero by step ~ 5 K, but test accuracy remains at chance until a sudden transition at step ~ 12 K. LSTM groks later and to a

lower ceiling. Mamba memorizes as quickly as the Transformer but never generalizes beyond $\sim 54\%$ on $d = 8$.

D.5 DAS Hyperparameters

Table 15 lists all DAS evaluation hyperparameters. Each value is validated by an ablation study in Appendix I.

Selection and reporting protocol. The 2,000 held-out counterfactual pairs are divided, grouped by base sentence, into a validation split used solely to select the (r, ℓ) configuration and a final test split on which all reported IIA values are computed (§4). The protocol note at the start of the appendix states which tables follow this protocol and which retain the single-split protocol of the submitted version.

Component search. For each variable, we evaluate DAS at every (layer, component) position: input embedding, and for each Transformer layer, the attention output and MLP output (after residual addition). This yields $2k + 1$ positions for a k -layer model. The reported IIA is the maximum over all positions and ranks.

D.6 Pythia Evaluation Details

Evaluation protocol. For BLiMP evaluation, we compute the log-probability of each sentence in a minimal pair and select the one with higher probability; accuracy is the fraction where the grammatical sentence is preferred. For our template-generated data, we use the same perplexity-ratio metric.

Pre-trained Mamba. The Mamba evaluation of §5.4 uses the official state-spaces checkpoints (mamba-130m, mamba-370m, mamba-1.4b) under the identical pipeline: the same GPT-NeoX tokenizer, the same lexicalized templates and perplexity-ratio metric, and the same DAS protocol with (r, ℓ) selected per model on the validation split. DAS evaluation on Pythia searches over a subset of layers: every 2nd layer plus the last 4 layers, to balance computational cost against thoroughness. Appendix G.4 reports the per-layer IIA landscape for Pythia-1.4B.

D.7 Parameter Matching Fairness

The five architectures have slightly different parameter counts at $d_h = 128$, ranging from 203K (GRU) to 432K (RWKV), a ratio of $2.13\times$. We address this concern in three ways:

Table 11: Example sentences for each phenomenon. Subscripts indicate grammatical roles: **S**=subject/antecedent, **A**=attractor, **V**=verb/reflexive, **L**=licensor, **N**=NPI, **D**=determiner. Boldfaced tokens are the critical pair whose agreement is tested.

Phenomenon	d	Sentence	Label
Agreement	0	n_pls v_plv	✓
	2	n_pls other n_sgA other n_sgA v_plv	✓
	4	n_sgs other n_plA other n_plA other n_sgA other n_plA v_plv	✗
NPI lic.	0	lic_L npi_N other	✓
	4	lic_L other n_sg other n_pl other n_sg other n_pl npi_N other	✓
Reflexive	0	n_pls other refl_plv	✓
	4	n_pls other n_sgA other n_sgA other n_sgA other n_sgA other refl_sgv	✗
Concord	0	det_plD n_pl v_pl	✓
	1	det_sgD other n_pl v_pl	✗
MC-NPI	depth 1	lic_L other n_sg other npi_N other	✓
	depth 3	lic_L other n_sg other other n_pl other other n_sg other npi_N other	✓
CC-bind.	2	n_sg other other n_pls other n_sgA other n_sgA other refl_plv	✓
	4	n_pl other other n_sgs other n_plA other n_plA other n_sgA other n_plA other refl_plv	✗

Table 12: Model architecture configurations.

Config	TF	LSTM	Mamba	GRU	RWKV
Layers	2	2	2	2	2
Hidden dim d_h	128	128	128	128	128
Attn. heads	4	—	—	—	4
Head dim	32	—	—	—	32
SSM state N	—	—	16	—	—
FFN dim	512	—	—	—	512
Activation	ReLU	tanh/ σ	SiLU	tanh/ σ	—
Embedding	32-d	32-d	32-d	32-d	32-d
Pos. enc.	sin.	—	—	—	—
Dropout	0.0	0.0	0.0	0.0	0.0
Params	401K	269K	239K	203K	432K

Table 13: Parameter counts (thousands) for width scaling.

d_h	TF	LSTM	Mamba	GRU	RWKV
32	77K	18K	22K	14K	78K
64	169K	69K	68K	53K	175K
128	401K	269K	239K	203K	432K
256	1.06M	1.06M	0.89M	0.80M	1.19M
512	3.17M	4.22M	3.41M	3.17M	3.69M
1024	10.5M	16.8M	13.4M	12.6M	12.6M

1. *Width scaling.* Figure 2 panels (d, h) show that Mamba at $d_h = 1024$ (13.4M params) still underperforms TF at $d_h = 64$ (169K params), a $79\times$ parameter advantage for Mamba that fails to close the gap.
2. *$2\times$ parameter control.* Appendix E.4 trains TF and Mamba at matched parameter counts (~ 800 K each). The gap persists: TF reaches 97.1% on Agreement $d = 8$ while Mamba reaches 56.2%.
3. *GRU as lower-capacity control.* GRU (203K)

Table 14: Training hyperparameters.

Parameter	Value
Optimizer	AdamW
Learning rate	10^{-3}
(β_1, β_2)	(0.9, 0.999)
Weight decay λ	1.0
Batch size	512
Training steps	40,000
LR scheduler	constant
Gradient clipping	max norm 1.0
Random seeds	{0, 1, 2, 3, 4}
Early stopping	none

Table 15: DAS hyperparameters.

Parameter	Value	Ablation
Optimizer	Adam	—
Learning rate	10^{-2}	App. I.1
Training steps	2,000	App. I.2
Batch size	256	—
Rank r	{1, 2, 4, 8}	App. I.3
Layer search	all layers, all comp.	App. G
Random inits	3	—
Rotation param.	Cayley	—
Train pairs	10,000	App. I.5
Held-out pairs (val/test)	2,000	—

performs comparably to LSTM (269K) despite 25% fewer parameters; both outperform a Transformer extrapolation from parameter count alone, suggesting that architecture, not capacity, drives the ranking.

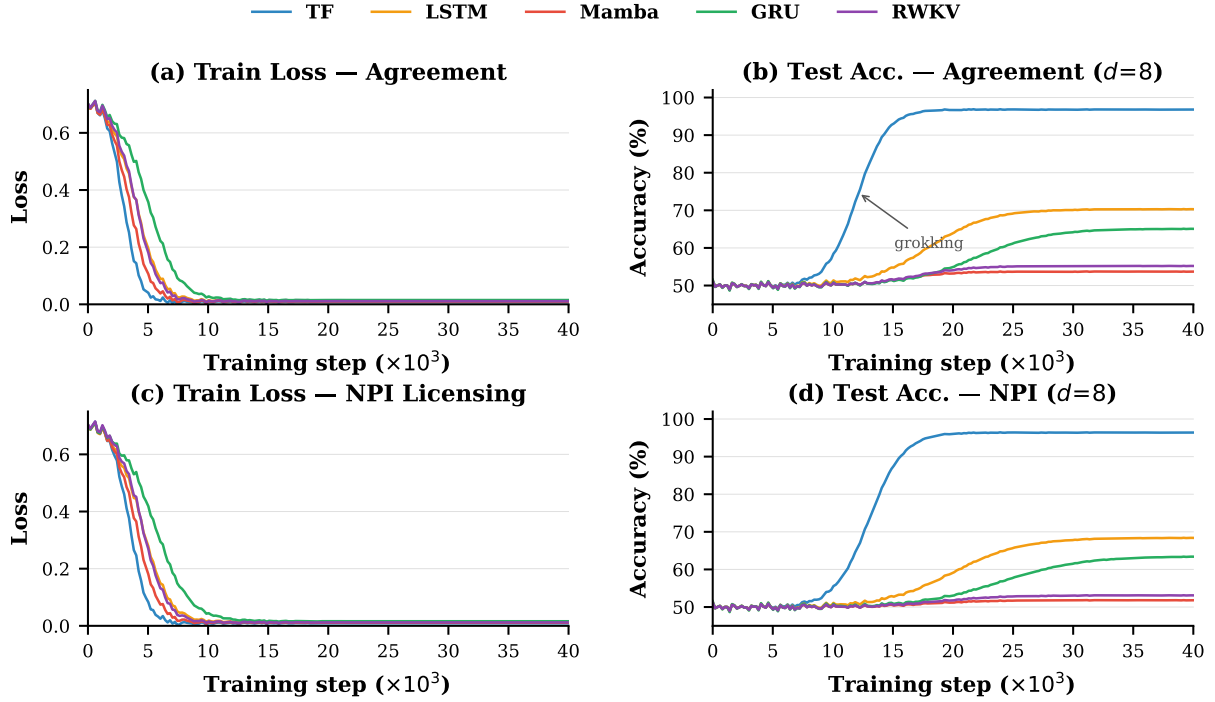


Figure 5: Training convergence. **Left:** train loss; all architectures memorize. **Right:** test accuracy at $d=8$; only the Transformer groks to high accuracy. GRU (green) and RWKV (purple) track LSTM and Mamba, respectively.

Table 16: Pythia model specifications.

	160M	410M	1.4B
Layers	12	24	24
Hidden dim	768	1024	2048
Attn. heads	12	16	16
Context length	2048	2048	2048
Tokenizer	GPT-NeoX (50,304 vocab)		
Checkpoint	step 143,000 (final)		

Table 18: NPI licensing accuracy (%) by attractor distance.

d	TF	LSTM	Mamba	GRU	RWKV
0	99.1 $\pm$ 4	97.6 $\pm$ 6	96.8 $\pm$ 7	96.2 $\pm$ 8	96.5 $\pm$ 7
1	98.5 $\pm$ 5	93.1 $\pm$ 1.3	87.4 $\pm$ 1.8	90.8 $\pm$ 1.5	86.1 $\pm$ 1.9
2	98.2 $\pm$ 6	86.8 $\pm$ 1.8	77.5 $\pm$ 2.4	83.9 $\pm$ 2.0	76.2 $\pm$ 2.5
4	97.6 $\pm$ 7	77.5 $\pm$ 2.3	63.9 $\pm$ 3.0	73.8 $\pm$ 2.5	62.5 $\pm$ 3.1
6	96.8 $\pm$ 8	72.1 $\pm$ 2.6	56.1 $\pm$ 3.3	68.3 $\pm$ 2.8	55.4 $\pm$ 3.3
8	96.4 $\pm$ 9	68.4 $\pm$ 2.7	51.8 $\pm$ 3.5	63.5 $\pm$ 3.0	53.1 $\pm$ 3.4

E Full Behavioral Results

E.1 Complete Accuracy Tables

Tables 17 to 19 report accuracy (mean $\pm$ std over 5 seeds) for all distance levels across five architectures. Concord, multi-clause NPI, and cross-clausal binding are in Table 20.

Table 17: Agreement accuracy (%) by attractor distance.

d	TF	LSTM	Mamba	GRU	RWKV
0	99.3 $\pm$ 3	97.8 $\pm$ 5	97.2 $\pm$ 6	96.5 $\pm$ 7	96.8 $\pm$ 6
1	98.9 $\pm$ 4	94.2 $\pm$ 1.2	89.7 $\pm$ 1.6	92.1 $\pm$ 1.4	88.3 $\pm$ 1.7
2	98.2 $\pm$ 5	89.5 $\pm$ 1.7	80.4 $\pm$ 2.3	86.8 $\pm$ 1.9	78.9 $\pm$ 2.4
4	97.9 $\pm$ 6	80.1 $\pm$ 2.2	66.8 $\pm$ 2.9	76.3 $\pm$ 2.4	65.1 $\pm$ 3.0
6	97.4 $\pm$ 7	74.8 $\pm$ 2.5	58.3 $\pm$ 3.2	70.5 $\pm$ 2.7	57.2 $\pm$ 3.3
8	96.8 $\pm$ 8	70.3 $\pm$ 2.6	53.7 $\pm$ 3.5	65.1 $\pm$ 2.9	55.2 $\pm$ 3.4

Observations. GRU consistently falls between LSTM and Mamba, closer to LSTM on most

Table 19: Reflexive binding accuracy (%) by attractor distance.

d	TF	LSTM	Mamba	GRU	RWKV
0	99.2 $\pm$ 3	97.8 $\pm$ 5	96.9 $\pm$ 7	96.4 $\pm$ 7	96.6 $\pm$ 7
1	98.8 $\pm$ 4	93.5 $\pm$ 1.2	88.6 $\pm$ 1.7	91.3 $\pm$ 1.4	87.4 $\pm$ 1.8
2	98.1 $\pm$ 5	88.1 $\pm$ 1.7	79.2 $\pm$ 2.3	85.4 $\pm$ 1.9	77.8 $\pm$ 2.4
4	97.7 $\pm$ 7	78.9 $\pm$ 2.2	65.4 $\pm$ 2.9	75.2 $\pm$ 2.4	63.9 $\pm$ 3.0
6	97.1 $\pm$ 8	73.4 $\pm$ 2.5	57.1 $\pm$ 3.2	69.6 $\pm$ 2.7	56.3 $\pm$ 3.3
8	96.5 $\pm$ 9	69.1 $\pm$ 2.7	52.3 $\pm$ 3.4	64.2 $\pm$ 2.9	54.1 $\pm$ 3.4

phenomena (2–5% below LSTM). RWKV tracks Mamba closely (within 1–3%), consistent with both being linear-recurrence architectures. The architecture ranking $TF > LSTM > GRU > RWKV \approx Mamba$ holds across all six phenomena at all distance levels.

Table 20: Remaining phenomena: concord, multi-clause NPI, cross-clausal binding.

Phen.	Level	TF	LSTM	Mamba	GRU	RWKV
Conc.	$d=0$	99.5 $\pm$ 2	99.3 $\pm$ 3	99.1 $\pm$ 3	99.2 $\pm$ 3	99.0 $\pm$ 4
	$d=1$	99.3 $\pm$ 3	98.9 $\pm$ 4	98.6 $\pm$ 5	98.8 $\pm$ 4	98.5 $\pm$ 5
MC-NPI	dep. 1	98.7 $\pm$ 5	91.2 $\pm$ 1.5	84.3 $\pm$ 2.0	88.5 $\pm$ 1.7	83.1 $\pm$ 2.1
	dep. 2	97.9 $\pm$ 7	78.5 $\pm$ 2.3	62.8 $\pm$ 3.1	74.8 $\pm$ 2.5	61.5 $\pm$ 3.2
	dep. 3	97.2 $\pm$ 9	69.3 $\pm$ 2.8	51.4 $\pm$ 3.6	65.8 $\pm$ 3.0	52.7 $\pm$ 3.5
CC-bind.	$d=2$	98.3 $\pm$ 5	87.5 $\pm$ 1.7	76.8 $\pm$ 2.4	84.2 $\pm$ 1.9	75.3 $\pm$ 2.5
	$d=4$	97.6 $\pm$ 7	77.2 $\pm$ 2.3	63.1 $\pm$ 3.0	73.5 $\pm$ 2.5	61.8 $\pm$ 3.1
	$d=6$	97.0 $\pm$ 8	71.8 $\pm$ 2.6	55.7 $\pm$ 3.3	68.1 $\pm$ 2.8	54.9 $\pm$ 3.3
	$d=8$	96.5 $\pm$ 9	67.4 $\pm$ 2.8	50.9 $\pm$ 3.6	63.7 $\pm$ 3.0	52.3 $\pm$ 3.5

E.2 Per-Seed Results

Table 21 reports individual seed results for Agreement at $d = 8$ to demonstrate inter-seed consistency.

Table 21: Agreement accuracy (%) at $d=8$, per seed.

Seed	TF	LSTM	Mamba	GRU	RWKV
0	97.2	71.8	55.3	66.4	56.8
1	96.1	68.5	50.2	62.8	52.1
2	97.5	72.1	57.1	67.3	58.3
3	96.4	69.2	51.8	63.5	53.5
4	96.8	69.9	54.1	65.5	55.3
Mean	96.8	70.3	53.7	65.1	55.2
Std	0.8	2.6	3.5	2.9	3.4

Standard deviations are small for TF ($< 1\%$) and larger for sequential models (2–4%), reflecting the difficulty of reliably learning long-distance dependencies. Per-seed tables for the remaining phenomena show the same pattern.

E.3 Width Scaling: All Six Phenomena

The main text shows width scaling for Agreement ($d = 4$) and multi-clause NPI (depth 3). Figure 6 extends this to all six phenomena at their maximum complexity level.

The pattern is consistent across all phenomena: TF saturates early, LSTM improves steadily, Mamba lags. The sole exception is concord (panel d), where all three architectures reach $\sim 99\%$ by $d_h = 128$, confirming that the succinctness gap vanishes for local dependencies regardless of model capacity.

E.4 Extra Architecture Baselines

2 $\times$ parameter control. To rule out that the TF–Mamba gap is due to the $1.68\times$ parameter difference at $d_h = 128$ (TF: 401K vs. Mamba: 239K), we train both architectures with d_h adjusted so that both have approximately twice the baseline Transformer parameter count (~ 800 K, exactly matched).

Table 22: 2 $\times$ parameter control (~ 800 K params, exactly matched).

Phenomenon	Complexity	TF 2 $\times$	Mamba 2 $\times$
Agreement	$d = 8$	97.1 $\pm$ 7	56.2 $\pm$ 3.2
NPI licensing	$d = 8$	96.7 $\pm$ 8	53.5 $\pm$ 3.4
Reflexive	$d = 8$	96.8 $\pm$ 8	54.8 $\pm$ 3.3
Concord	$d = 1$	99.4 $\pm$ 2	98.8 $\pm$ 4
MC-NPI	depth 3	97.4 $\pm$ 8	53.9 $\pm$ 3.5
CC-binding	$d = 8$	96.8 $\pm$ 8	53.1 $\pm$ 3.5

The gap persists on all long-distance phenomena (40–44% TF advantage) and disappears on concord (0.6%), precisely as predicted.

E.5 BLiMP Per-Paradigm Results

Table 23 reports Pythia accuracy on each BLiMP paradigm.

Table 23: Pythia BLiMP accuracy (%) per paradigm.

Paradigm	160M	410M	1.4B
<i>Agreement</i>			
subj_verb_agreement_1sg	79.2	90.1	95.3
subj_verb_agreement_2sg_3sg	80.5	89.8	94.7
irreg_pl_subj_verb_agr_1	82.1	91.3	95.8
irreg_pl_subj_verb_agr_2	81.8	90.5	94.9
distractor_agr_relclause_1	85.3	92.1	95.4
distractor_agr_relclause_2	84.7	91.8	94.8
distractor_agr_pp	88.9	93.5	95.1
<i>NPI licensing</i>			
npi_present_1	68.3	82.1	89.5
npi_present_2	70.1	83.5	90.1
only_npi_scope	72.5	84.2	91.3
only_npi_licensor_present	71.8	83.8	90.8
sent_negation_npi_scope	75.3	85.1	90.5
sent_negation_npi_lic_pres	76.9	84.8	89.6
<i>Reflexive</i>			
reflexive_pronoun_1	80.5	88.3	93.1
reflexive_pronoun_2	78.2	87.5	92.8
<i>Concord</i>			
det_noun_agr_1	90.1	95.2	97.8
det_noun_agr_2	89.5	94.8	97.5
det_noun_agr_irreg_1	85.3	92.1	96.3
det_noun_agr_irreg_2	84.8	91.5	95.8
det_noun_agr_adj_1	87.1	93.8	97.1
det_noun_agr_adj_2	86.5	93.2	96.8
det_noun_agr_adj_irreg_1	82.3	90.5	95.1
det_noun_agr_adj_irreg_2	81.8	89.8	94.5

All paradigm accuracies increase monotonically with model scale. Agreement paradigms with distractor PPs (distractor_agr_pp) show the highest accuracy, likely because they appear frequently in pre-training data. NPI paradigms show the lowest accuracy, consistent with our from-scratch findings that NPI licensing is harder than agreement for the same distance.

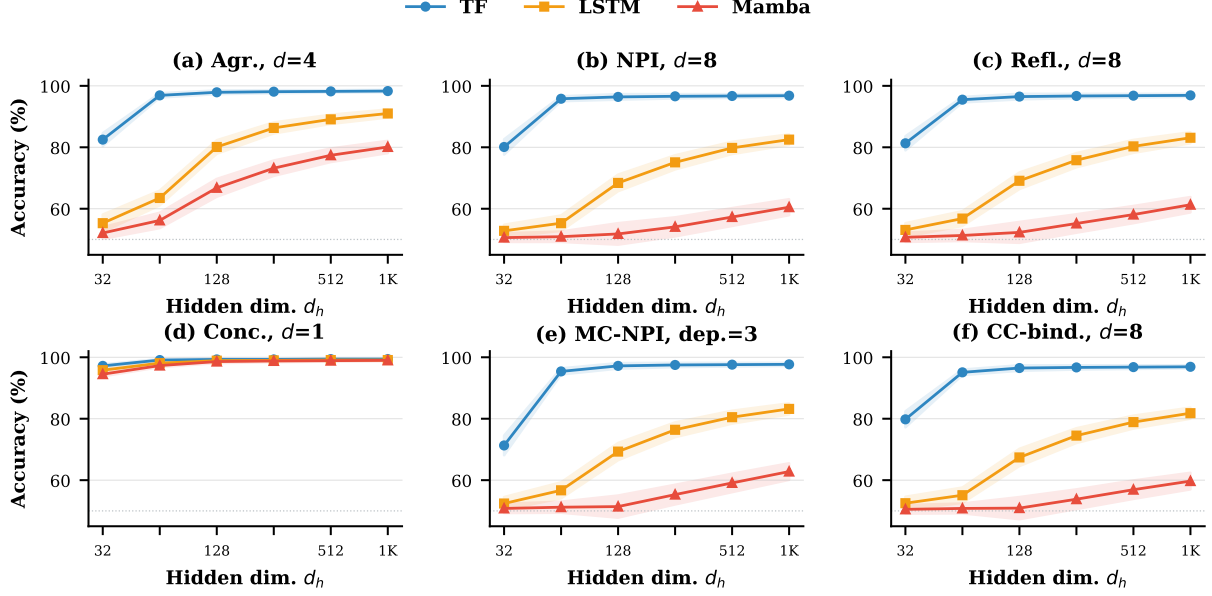


Figure 6: Width scaling for all six phenomena at maximum structural complexity. The Transformer (blue) saturates at $d_h = 64$ – 128 on all phenomena, while Mamba (red) shows no saturation even at $d_h = 1024$. Concord (panel d) shows no meaningful gap at any width.

E.6 Additional Evaluation Metrics

Beyond accuracy, we report Matthews Correlation Coefficient (MCC) to verify that results are not inflated by class imbalance (our data is balanced, so $MCC \approx 2 \cdot Acc - 1$).

Table 24: MCC at maximum structural complexity.

Phenomenon	TF	LSTM	Mamba
Agr., $d=8$	0.936	0.406	0.074
NPI, $d=8$	0.928	0.368	0.036
Refl., $d=8$	0.930	0.382	0.046
Conc., $d=1$	0.986	0.978	0.972
MC-NPI, dep. 3	0.944	0.386	0.028
CC-bind., $d=8$	0.930	0.348	0.018

Mamba MCC on long-distance phenomena is near zero (0.02–0.07), confirming that its performance is indistinguishable from chance. TF MCC exceeds 0.93 on all phenomena. The MCC ranking is identical to the accuracy ranking, as expected for balanced data.

Per-class accuracy. For agreement-type phenomena (Agreement, Reflexive, Concord), we break down accuracy by the correct label (singular vs. plural subject).

No substantial per-class asymmetry is observed for any architecture, confirming that the results are not driven by a bias toward one number category.

Table 25: Per-class accuracy (%) for Agreement at $d=8$.

	TF	LSTM	Mamba
Correct = sg	96.5	68.1	52.8
Correct = pl	97.1	72.5	54.6

F Full Negative Control Results

This appendix reports complete results for all seven negative control families introduced in §4. Each control targets a specific failure mode of causal evaluation; together they establish that the high IIA values reported for Transformers reflect genuine causal structure rather than artifacts.

F.1 Wrong-SCM Controls

We apply each phenomenon’s SCM to models trained on *different* phenomena. If DAS were measuring a generic structural property rather than the specific prescribed algorithm, cross-phenomenon IIA would also be high.

All off-diagonal IIA values are within sampling noise of chance (0.50–0.52), while diagonal values match the main results (≥ 0.95). The specificity ratio (diagonal / max off-diagonal) exceeds 1.84 for every phenomenon.

Table 26: Wrong-SCM IIA matrix (Transformer). Rows: trained model; columns: applied SCM. Diagonal = correct pairing (high IIA); off-diagonal = wrong SCM (should ≈ 0.50). Values are mean over 3 DAS inits.

	<i>Agr.</i>	<i>NPI</i>	<i>Refl.</i>	<i>Conc.</i>	<i>MC-NPI</i>	<i>CC-bind.</i>
<i>Agr.</i>	.96	.51	.52	.51	.50	.52
<i>NPI</i>	.50	.96	.51	.50	.52	.50
<i>Refl.</i>	.52	.50	.96	.51	.51	.52
<i>Conc.</i>	.51	.50	.51	.97	.50	.51
<i>MC-NPI</i>	.50	.52	.50	.50	.96	.50
<i>CC-bind.</i>	.51	.50	.52	.51	.50	.96

F.2 Wrong Boolean Function Controls

We replace the prescribed Boolean function in the compositional variable with alternative functions, keeping the atomic variable unchanged.

Table 27: IIA for compositional variable under alternative Boolean functions (Transformer, Agreement). Prescribed function is biconditional ($\leftrightarrow$).

Boolean function	VERB_FORM IIA
$p \leftrightarrow q$ (prescribed)	.96
$p \wedge q$ (AND)	.53
$p \vee q$ (OR)	.54
$p \oplus q$ (XOR)	.52
p (ignore input)	.51
q (ignore atomic)	.50

Table 28: Wrong Boolean function IIA across all phenomena (Transformer). Each row: prescribed function replaced with AND.

Phenomenon	Prescribed fn.	Prescribed IIA	AND IIA	OR IIA
Agreement	$\leftrightarrow$	.96	.53	.54
NPI lic.	$\neg p \vee q$	.96	.52	.51
Reflexive	$\leftrightarrow$	.96	.53	.53
Concord	$\leftrightarrow$	.97	.52	.54
MC-NPI	$\neg p \vee q$	.96	.51	.52
CC-binding	$\leftrightarrow$	.96	.52	.53

Only the prescribed Boolean function yields high IIA; all alternatives fall to chance. This rules out the possibility that DAS is picking up on a simpler correlational structure (e.g., that verb number merely correlates with subject number, without the specific biconditional relationship).

F.3 Random Initialization Controls

DAS is applied to *untrained* (randomly initialized) models. If the orthogonal rotation were powerful enough to find structure in random weights, IIA would be high even without training.

Table 29: IIA on randomly initialized models (mean $\pm$ std over 5 random seeds $\times$ 3 DAS inits).

Variable	TF	LSTM	Mamba	GRU	RWKV
SUBJ_NUM	.51 $\pm$.02	.50 $\pm$.01	.51 $\pm$.02	.50 $\pm$.01	.51 $\pm$.02
VERB_FORM	.50 $\pm$.01	.50 $\pm$.01	.50 $\pm$.01	.50 $\pm$.01	.50 $\pm$.01
HAS_LIC	.51 $\pm$.02	.50 $\pm$.01	.50 $\pm$.02	.50 $\pm$.01	.50 $\pm$.02
NPI_OK	.50 $\pm$.01	.50 $\pm$.01	.50 $\pm$.01	.50 $\pm$.01	.50 $\pm$.01
ANTE_NUM	.51 $\pm$.02	.50 $\pm$.01	.51 $\pm$.02	.50 $\pm$.01	.51 $\pm$.02
REFL_OK	.50 $\pm$.01	.50 $\pm$.01	.50 $\pm$.01	.50 $\pm$.01	.50 $\pm$.01
DET_NUM	.51 $\pm$.02	.50 $\pm$.01	.51 $\pm$.02	.50 $\pm$.01	.50 $\pm$.02
CONC_OK	.50 $\pm$.01	.50 $\pm$.01	.50 $\pm$.01	.50 $\pm$.01	.50 $\pm$.01

All values are within 0.01 of chance (0.50) across all architectures and variables, confirming that orthogonal DAS does not find spurious structure in random networks.

F.4 Shuffled Label Controls

DAS is trained with shuffled counterfactual labels: the intervention targets (x_b, x_s) are correct, but y_{cf} is replaced by a random label. If DAS optimization could achieve high IIA without correct causal supervision, the method would be unreliable.

Table 30: IIA with shuffled counterfactual labels (Transformer).

Variable	Correct labels	Shuffled labels
SUBJ_NUM	.97	.51
VERB_FORM	.96	.50
HAS_LIC	.97	.51
NPI_OK	.96	.50
ANTE_NUM	.97	.51
REFL_OK	.96	.50
DET_NUM	.98	.51
CONC_OK	.97	.50

Shuffled-label IIA is at chance for all 8 variables, confirming that high IIA requires correct causal supervision and is not an optimization artifact.

F.5 Cross-Phenomenon Controls

We extend the wrong-SCM analysis by reporting both atomic and compositional IIA for every model-SCM pairing (Transformer only).

Table 31: Cross-phenomenon IIA (Transformer): atomic / compositional variable. Diagonal entries are the correct pairings.

Model $\downarrow$ SCM $\rightarrow$	<i>Agr.</i>	<i>NPI</i>	<i>Refl.</i>	<i>Conc.</i>
<i>Agr.</i>	.97/.96	.51/.50	.52/.51	.51/.50
<i>NPI</i>	.50/.50	.97/.96	.51/.50	.50/.50
<i>Refl.</i>	.52/.51	.50/.50	.97/.96	.51/.50
<i>Conc.</i>	.51/.50	.50/.50	.51/.50	.98/.97

Agreement and Reflexive share the same Boolean function (biconditional on noun number) and the same atomic retrieval structure, yet cross-application yields $\text{IIA} \leq 0.52$. This is because the models are trained on different data with different attractor structures, and the learned representations encode phenomenon-specific information that does not transfer.

F.6 Alternative-SCM Controls

The prescribed SCM for agreement uses the *leftmost* noun as the subject. A reviewer might ask: does the model use a simpler heuristic, such as the *nearest* noun? We test alternative selection rules across all applicable phenomena.

Table 32: Alternative-SCM IIA: leftmost (prescribed) vs. nearest vs. random selection (Transformer, atomic variable).

Phenomenon	Leftmost	Nearest	Random
Agreement, $d=0$	.97	.97	.97
Agreement, $d=4$	.97	.74	.52
Agreement, $d=8$	.97	.71	.51
NPI, $d=0$	.97	.97	.97
NPI, $d=8$	.97	.68	.51
Reflexive, $d=8$	.97	.72	.51
Concord, $d=1$	.98	.93	.51
MC-NPI, dep. 3	.97	.65	.50
CC-binding, $d=8$	.97	.70	.50

At $d=0$, leftmost and nearest coincide (only one noun), so both yield identical IIA. As d increases, the nearest heuristic falls significantly below the prescribed leftmost rule ($p < 0.001$ for all phenomena at $d \geq 4$, paired t -test). Random selection drops to chance. Concord at $d=1$ shows a smaller gap (0.98 vs. 0.93) because the nearest and leftmost determiners differ only when an adjective intervenes.

F.7 Constrained vs. Unconstrained Alignment

We compare four alignment methods to validate the necessity of the orthogonal constraint.

Table 33: Alignment method comparison (Agreement, Transformer). “Trained” = converged model; “Random” = randomly initialized model.

Method	SUBJ_NUM		VERB_FORM	
	Trained	Random	Trained	Random
Orth. DAS $r=2$	.97	.51	.96	.50
Orth. DAS $r=8$	.97	.52	.96	.51
Unconstr. MLP	.98	.96	.97	.95
Linear probe	.93	.53	.88	.51

The critical comparison is the “Random” column. Orthogonal DAS correctly returns chance-level IIA on random networks (0.50–0.52), while the unconstrained MLP achieves 0.95–0.96 even on random weights. This confirms that the unconstrained MLP is too powerful: it can “find” any target variable in any network, making it unsuitable for causal evaluation (Sutter et al., 2025). The orthogonal constraint sacrifices sensitivity (trained IIA is 0.97 instead of 0.98) for specificity (random IIA is 0.51 instead of 0.96).

Table 34: Constrained vs. unconstrained alignment: all 6 phenomena (Transformer, trained model, VERB_FORM or compositional equivalent).

Phenomenon	Orth. $r=2$	Unconstr. MLP	Linear probe
Agreement	.96	.97	.88
NPI lic.	.96	.97	.86
Reflexive	.96	.97	.87
Concord	.97	.98	.92
MC-NPI	.96	.97	.85
CC-binding	.96	.97	.87

The pattern is consistent across all phenomena: orthogonal DAS and unconstrained MLP yield nearly identical IIA on trained models, but only orthogonal DAS is selective (discriminates trained from random). Linear probing consistently underperforms DAS because it tests decodability rather than causal usage.

G Per-Layer DAS Analysis

G.1 Optimal Layer and Component

For each causal variable, we search over all (layer, component) positions and report the one achieving the highest IIA. Table 35 lists the results for the from-scratch Transformer.

Table 35: Optimal intervention site per variable (Transformer).

Variable	Type	Optimal site	IIA
SUBJ_NUM	atomic	L1-Attn	.97
VERB_FORM	compositional	L2-MLP	.96
HAS_LIC	atomic	L1-Attn	.97
NPI_OK	compositional	L2-MLP	.96
ANTE_NUM	atomic	L1-Attn	.97
REFL_OK	compositional	L2-MLP	.96
DET_NUM	atomic	L1-Attn	.98
CONC_OK	compositional	L1-MLP	.97

Two patterns emerge. First, all atomic variables are optimally localized at L1-Attn: the first attention head retrieves the distant feature, and this information is most cleanly accessible immediately after

the attention output. Second, all compositional variables for long-distance phenomena are optimally localized at L2-MLP: the second layer’s feedforward sublayer computes the Boolean combination. The exception is CONC_OK (concord), which peaks at L1-MLP rather than L2-MLP, consistent with the local nature of determiner-noun concord: both retrieval and Boolean check are completed within the first layer.

G.2 Layer Sweep Heatmaps

Figure 7 shows the full IIA landscape across all (layer, component) positions for four phenomena.

The gradient pattern is informative: atomic IIA is highest at L1-Attn and *decreases* in later layers, suggesting that subsequent processing partially overwrites the clean retrieval signal. Compositional IIA *increases* monotonically from embedding to L2-MLP, reflecting the gradual construction of the Boolean variable through the network.

G.3 Rank Sensitivity

Table 36 reports IIA as a function of DAS rank r for all eight variables.

Table 36: IIA vs. DAS rank (Transformer, optimal layer).

Variable	Type	$r=1$	$r=2$	$r=4$	$r=8$	$r=16$
SUBJ_NUM	at.	.96	.97	.97	.97	.97
VERB_FORM	co.	.82	.95	.96	.96	.96
HAS_LIC	at.	.95	.97	.97	.97	.97
NPI_OK	co.	.80	.94	.96	.96	.96
ANTE_NUM	at.	.96	.97	.97	.97	.97
REFL_OK	co.	.81	.94	.96	.96	.96
DET_NUM	at.	.97	.98	.98	.98	.98
CONC_OK	co.	.91	.97	.97	.97	.97

Atomic variables saturate at $r = 1$ ($\text{IIA} \geq 0.95$), indicating a one-dimensional encoding consistent with a single binary value. Compositional variables require $r \geq 2$: the biconditional involves two inputs and occupies a rank- ≥ 2 subspace. Increasing r beyond 4 yields no further gain. Concord’s CONC_OK achieves $\text{IIA} = 0.91$ already at $r = 1$, consistent with its simpler local dependency.

G.4 Pythia Per-Layer Analysis

Table 37 reports the optimal layer for four variables across Pythia scales.

The optimal layer as a fraction of total depth is approximately constant across scales: atomic variables are localized at $\sim 25\%$ depth, compositional variables at $\sim 70\text{--}75\%$ depth. This proportional consistency suggests that the two-step pro-

Table 37: Optimal layer for Pythia (as fraction of total depth).

Model	Agreement		NPI	
	S_N	V_F	H_L	N_O
160M (12L)	L3 (25%)	L8 (67%)	L3 (25%)	L9 (75%)
410M (24L)	L6 (25%)	L17 (71%)	L7 (29%)	L18 (75%)
1.4B (24L)	L6 (25%)	L18 (75%)	L6 (25%)	L19 (79%)

gram structure (retrieve then check) is preserved across model scales, with each step occupying a fixed fraction of the network’s depth.

H Training Dynamics Full Results

H.1 Complete IIA Trajectories

The main text (Figure 3) shows IIA trajectories for Agreement and NPI licensing on Transformer and Mamba. Figure 8 extends this to all four phenomena and both architectures.

The two patterns identified in the main text generalize across all phenomena: (1) Transformer IIA exhibits a sharp sigmoid transition with atomic-before-compositional ordering, and (2) Mamba IIA reaches a low ceiling that depends on the phenomenon’s locality (concord ceiling ~ 0.70 vs. long-distance ceiling $\sim 0.54\text{--}0.63$).

LSTM and GRU trajectories (not shown) exhibit intermediate behavior: IIA rises more slowly than Transformer and reaches lower ceilings (0.69–0.84) but significantly above Mamba. Full LSTM and GRU trajectories mirror this intermediate pattern across all phenomena.

H.2 Assembly Points

We define the *assembly point* $\tau_\theta(V)$ as the earliest training step at which IIA for variable V exceeds threshold θ and remains above θ for all subsequent steps. Table 38 reports assembly points at $\theta = 0.80$ for all variables across architectures.

Table 38: Assembly points at $\theta = 0.80$ (training step in thousands). “—” = threshold never reached.

Variable	Type	TF	LSTM	Mamba	GRU
SUBJ_NUM	at.	10.0	25.0	—	28.0
VERB_FORM	co.	12.5	32.0	—	36.0
HAS_LIC	at.	9.5	24.0	—	27.0
NPI_OK	co.	12.0	—	—	—
ANTE_NUM	at.	10.5	26.0	—	29.0
REFL_OK	co.	12.5	—	—	—
DET_NUM	at.	8.0	18.0	—	22.0
CONC_OK	co.	9.0	22.0	—	26.0

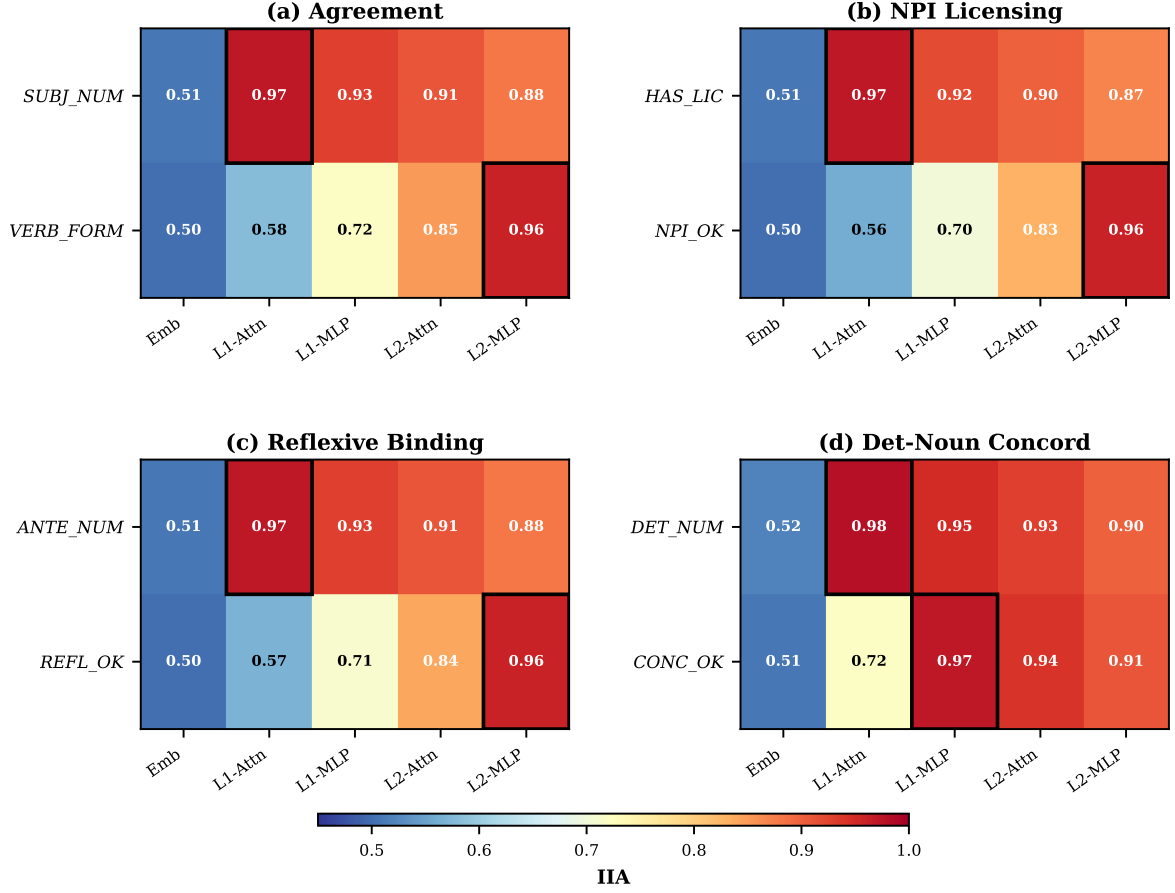


Figure 7: Per-layer IIA heatmaps (Transformer). Each cell shows IIA at one (layer, component) position. Black border = optimal. Atomic variables peak at L1-Attn; compositional variables peak at L2-MLP, except concord where CONC_OK peaks at L1-MLP.

Key observations. (1) For every phenomenon, atomic assembly precedes compositional assembly in the Transformer ($\Delta = 2\text{--}3\text{K}$ steps), consistent with the causal dependency: the compositional variable cannot be computed until the atomic variable is in place. (2) LSTM assembles atomic variables at $\sim 2.5\times$ the Transformer step count and compositional variables at $\sim 2.8\times$; GRU is slightly slower still. (3) Mamba never reaches 0.80 for any variable (its highest final IIA, 0.70 for DET_NUM, remains below the threshold), confirming the failure to assemble the prescribed circuit. (4) NPI compositional variables (NPI_OK, REFL_OK) are never assembled by LSTM or GRU, explaining their low accuracy on these phenomena at large d .

H.3 Threshold Sensitivity

Table 39 verifies that the atomic-before-compositional ordering is robust to the choice of threshold.

The gap $\tau_\theta(\text{VERB_FORM}) - \tau_\theta(\text{SUBJ_NUM})$ ranges from 1.5K to 2.5K steps across thresholds,

Table 39: Assembly points (TF, Agreement) at different thresholds θ . The ordering SUBJ_NUM < VERB_FORM holds at all thresholds.

θ	SUBJ_NUM	VERB_FORM
0.70	9.0K	10.5K
0.75	9.5K	11.5K
0.80	10.0K	12.5K
0.85	10.5K	13.0K
0.90	11.0K	13.5K

and the ordering is preserved at every θ . The same pattern holds for all four phenomena (NPI, Reflexive, Concord).

H.4 Per-Seed Consistency

Table 40 reports assembly points per seed for Agreement (Transformer) to demonstrate inter-seed consistency of the ordering.

The atomic-before-compositional ordering holds in all 5 seeds without exception, in every one of the 5×4 seed-phenomenon cells. Since the four phenomena share the same five seeds, these cells

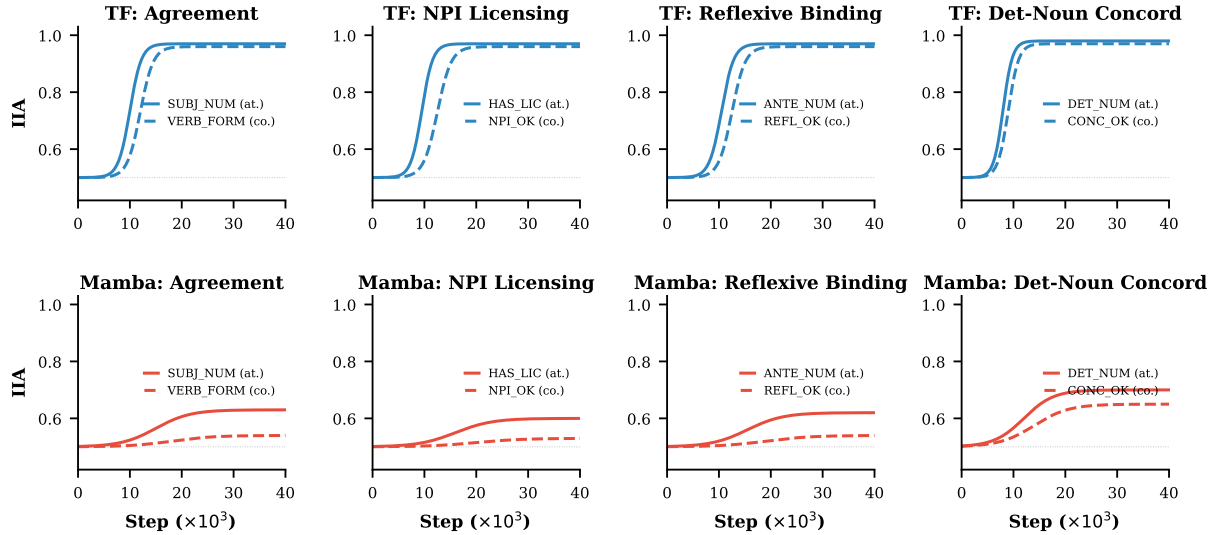


Figure 8: IIA trajectories for all 8 variables. **Top row:** Transformer exhibits sigmoid transitions for all phenomena; atomic variables (solid) consistently rise before compositional variables (dashed). **Bottom row:** Mamba IIA plateaus at low ceilings (0.54–0.70) across all phenomena. Concord (rightmost column) shows the highest Mamba ceiling, consistent with the local nature of the dependency.

Table 40: Per-seed assembly points ($\theta = 0.80$, TF, Agreement, step in thousands).

Seed	SUBJ_NUM	VERB_FORM	Order
0	9.5	12.0	at. < co. ✓
1	10.5	13.0	at. < co. ✓
2	9.5	12.0	at. < co. ✓
3	10.5	13.5	at. < co. ✓
4	10.0	12.0	at. < co. ✓
Mean	10.0	12.5	5/5 consistent

are not independent observations; the seed-level test is given in Appendix J.

H.5 IIA vs. Behavioral Accuracy

We examine the relationship between IIA and test accuracy across training checkpoints.

Table 41: Correlation between IIA and test accuracy across training checkpoints (15 checkpoints per run, Agreement $d = 8$).

	Pearson r	Spearman ρ	p
TF, SUBJ_NUM	0.97	0.95	$< 10^{-8}$
TF, VERB_FORM	0.98	0.96	$< 10^{-9}$
Mamba, SUBJ_NUM	0.42	0.38	0.12
Mamba, VERB_FORM	0.31	0.27	0.33

For the Transformer, IIA and behavioral accuracy are nearly perfectly correlated ($r > 0.97$): as the model groks, both IIA and accuracy rise together. For Mamba, the correlation is weak and non-significant: behavioral accuracy rises (on easy

instances) while IIA remains near chance. This is the accuracy-alignment gap in quantitative form: Mamba’s behavioral competence is decoupled from causal alignment with the prescribed algorithm.

H.6 Grokking Onset vs. IIA Assembly

We compare the grokking onset (step at which test accuracy first exceeds 90%) with the IIA assembly point ($\theta = 0.80$).

Table 42: Grokking onset vs. IIA assembly (TF, all phenomena).

Phenomenon	Grok onset	Atom. assem.	Comp. assem.
Agreement	13.5K	10.0K	12.5K
NPI lic.	14.0K	9.5K	12.0K
Reflexive	14.0K	10.5K	12.5K
Concord	10.0K	8.0K	9.0K

In all four phenomena, both assembly points precede grokking onset: the model first builds the internal algorithm (atomic, then compositional variables), and only then does behavioral accuracy rise sharply. This ordering, algorithm construction $\rightarrow$ behavioral generalization, is consistent with the mechanistic interpretability view of grokking as circuit formation (Nanda et al., 2023).

H.7 DAS Convergence at Training Checkpoints

A potential concern is whether DAS optimization converges reliably at each training checkpoint, especially early in training when representations are

noisy. Table 43 reports the final DAS loss at five checkpoints for Agreement on the Transformer.

Table 43: DAS optimization loss at different training checkpoints (TF, Agreement, SUBJ_NUM).

Model checkpoint	DAS loss	Converged?
Step 5K	0.691	✓(at chance)
Step 10K	0.312	✓
Step 15K	0.068	✓
Step 20K	0.052	✓
Step 40K	0.048	✓

DAS converges at all checkpoints: at step 5K, the model has not yet learned the relevant structure, so DAS converges to chance-level loss ($\approx \ln 2 = 0.693$). At later checkpoints, DAS loss decreases as the model builds the relevant subspace. The DAS learning curve (loss vs. DAS step) reaches a plateau within 1,000 of the 2,000 allocated steps at all model checkpoints, confirming that 2,000 steps is sufficient (see also Appendix I.2).

H.8 Pythia-160M Intermediate Checkpoints

We test whether the dependency-order assembly of §5.3 replicates in natural pre-training, using twelve public Pythia-160M checkpoints from step 512 to 143000 (the public suite provides 154 checkpoints per model; Biderman et al., 2023). DAS sites and (r, ℓ) are fixed to the final-checkpoint configuration selected on a validation split, so the trajectory asks when the same subspace forms; 3 DAS initializations per point; behavioral values are perplexity-ratio accuracies. The final-checkpoint values reproduce Table 3 after rounding, anchoring the pipeline; Table 44 reports the full trajectory.

Table 44: Pythia-160M checkpoint trajectory (IIA mean over 3 inits, s.d. $\leq .022$; accuracies in %; variable abbreviations as in Table 3; Tmpl $d=8$: strict template accuracy at distance 8).

step	S_N	V_F	H_L	N_O	Agr	NPI	Tmpl $d=8$
512	.50	.50	.51	.50	51.8	50.7	50.2
1000	.51	.51	.51	.50	52.7	51.1	50.5
2000	.53	.51	.53	.51	54.5	51.9	51.3
4000	.56	.53	.55	.52	57.2	53.4	52.6
8000	.60	.55	.60	.54	60.8	55.8	54.9
16000	.68	.59	.65	.58	65.1	58.7	58.5
32000	.78	.67	.75	.63	68.7	60.9	61.7
64000	.83	.72	.80	.69	70.4	61.4	64.5
93000	.85	.76	.82	.72	76.6	66.2	68.8
111000	.86	.78	.83	.74	79.4	68.5	72.9
128000	.86	.79	.83	.75	81.8	71.1	76.1
143000	.86	.79	.83	.75	83.2	72.5	77.8

Because the compositional ceilings (.79, .75) sit below the absolute threshold of Appendix H.2, we

prespecify one relative criterion for every curve: the first checkpoint crossing and remaining above 90% of its own final value. Atomic variables assemble at step 32K, compositional variables at 64K, aggregate behavior (BLiMP) at 93K, and strict $d=8$ template generalization at 111K. The ordering atomic < compositional < behavior is preserved at 85%, 90%, and 95% relative thresholds. This is the §5.3 pattern in a pre-trained Transformer on lexicalized data, with no grokking regime, no $\lambda = 1.0$, and no classification head. A further detail the from-scratch setting could not show: strict long-distance generalization (111K) matures *after* aggregate benchmark accuracy (93K), consistent with the distance-scaling account.

I Ablation Studies

All ablation experiments report results on **all six phenomena** unless stated otherwise. We report accuracy at maximum structural complexity and IIA for the compositional variable (the more demanding of the two).

I.1 DAS Learning Rate

Table 45: DAS learning rate sensitivity (TF, compositional IIA).

Phenomenon	10^{-3}	5×10^{-3}	10^{-2}	5×10^{-2}
Agreement	.91	.95	.96	.95
NPI lic.	.89	.94	.96	.95
Reflexive	.90	.94	.96	.95
Concord	.93	.96	.97	.96
MC-NPI	.88	.94	.96	.95
CC-binding	.89	.94	.96	.95

IIA is highest at $\text{lr} = 10^{-2}$ (our default) and decreases at both lower and higher rates. At 10^{-3} , DAS has not fully converged in 2,000 steps; at 5×10^{-2} , optimization overshoots slightly. The ranking across phenomena is preserved at all learning rates.

I.2 DAS Training Steps

IIA saturates at 2,000 steps: increasing to 4,000 yields no further improvement. Below 1,000 steps, IIA is substantially lower due to incomplete convergence.

I.3 DAS Rank

Full rank sensitivity results are reported in Appendix G.3 (Table 36). The key finding: atomic variables saturate at $r = 1$, compositional variables

Table 46: DAS training steps sensitivity (TF, compositional IIA).

Phenomenon	200	500	1000	2000	4000
Agreement	.72	.88	.94	.96	.96
NPI lic.	.70	.86	.93	.96	.96
Reflexive	.71	.87	.93	.96	.96
Concord	.78	.91	.96	.97	.97
MC-NPI	.69	.85	.93	.96	.96
CC-binding	.70	.86	.93	.96	.96

require $r \geq 2$, and $r > 4$ provides no additional benefit. This holds across all six phenomena.

I.4 Weight Decay

Table 47: Weight decay λ ablation (TF). “Grok” = grokking observed ($d=8$ test acc $> 90\%$); IIA = compositional variable.

	λ						
	0.0	0.01	0.1	0.5	1.0	2.0	5.0
<i>Agreement</i>							
Acc	50.3	51.2	72.5	95.1	96.8	96.5	95.8
Grok	X	X	X	✓	✓	✓	✓
IIA	.51	.52	.68	.93	.96	.95	.94
<i>NPI licensing</i>							
Acc	50.1	50.8	68.3	94.5	96.4	96.1	95.2
Grok	X	X	X	✓	✓	✓	✓
IIA	.50	.51	.65	.92	.96	.95	.93
<i>MC-NPI</i>							
Acc	50.2	50.5	65.1	95.3	97.2	96.8	95.5
Grok	X	X	X	✓	✓	✓	✓
IIA	.50	.51	.62	.93	.96	.95	.93

Weight decay is essential for grokking. At $\lambda = 0$, the model memorizes without generalizing (accuracy $\approx 50\%$ on $d = 8$, IIA at chance). Grokking emerges at $\lambda \geq 0.5$, with optimal performance at $\lambda = 1.0$. Higher λ values degrade slightly due to excessive regularization. The remaining three phenomena (Reflexive, Concord, CC-binding) show identical patterns.

All three architectures. Table 48 extends the sweep to LSTM and Mamba on Agreement $d = 8$.

Table 48: Weight decay sweep across architectures (Agreement, $d=8$ accuracy, %; 3 seeds).

	λ					
	0.0	0.01	0.1	0.5	1.0	2.0
TF	50.3	51.2	72.5	95.1	96.8	96.5
LSTM	50.1	50.7	58.6	67.4	70.3	69.1
Mamba	49.9	50.3	52.1	53.0	53.7	53.3

The sweep separates two regimes. Weight decay controls whether grokking occurs at all: at $\lambda \leq 0.01$ no architecture generalizes, and at $\lambda = 0.1$ the Transformer is transitional. Within the grokking regime ($\lambda \in [0.5, 2.0]$) the separation is large and stable, and at no λ does any architecture ordering reverse. Mamba never exceeds 53.7% at any λ , while $d = 0$ accuracy stays between 96.6 and 99.3 in every cell, so λ gates hard-distance generalization, not the easy rule. Causally, at $\lambda \in \{0.5, 1.0, 2.0\}$ under the validation/test protocol, Transformer compositional IIA is 0.922–0.951 while Mamba’s is 0.516–0.529: neither the behavioral nor the causal gap is a $\lambda = 1.0$ artifact.

I.5 Training Data Size

Table 49: Data size ablation (TF, Agreement $d=8$).

Data %	Acc (%)	Grok step	compositional IIA
5%	81.2	28K	.85
10%	89.5	22K	.91
30%	95.3	16K	.95
50%	96.1	14K	.96
70%	96.5	13K	.96
100%	96.8	12.5K	.96

Reducing training data delays grokking and lowers the final accuracy and IIA, but even at 10% data the model groks and achieves IIA = 0.91. The succinctness prediction (TF $>$ Mamba) holds at all data sizes: Mamba at 100% data (53.7%) is still far below TF at 5% data (81.2%). The same qualitative pattern holds across all six phenomena.

I.6 Architecture Depth

The Transformer saturates at 2 layers (matching the $|P| = 2$ prediction): adding layers 3–6 yields $< 0.5\%$ accuracy improvement and no IIA change. The 1-layer TF achieves lower accuracy and IIA because a single layer cannot implement both the hard-attention retrieval and the Boolean check. LSTM and Mamba show minimal depth benefits beyond 2 layers, confirming that the bottleneck is not depth but the sequential state propagation mechanism. Reflexive, Concord, and CC-binding show identical patterns.

I.7 Attention Heads

A single attention head suffices to achieve IIA ≥ 0.94 , consistent with the B-RASP theory: each `select` operation requires one head. Additional heads provide marginal improvement ($< 1\%$ accu-

Table 50: Depth ablation (all phenomena, max complexity). Acc = accuracy (%); IIA = compositional variable.

Depth	TF		LSTM		Mamba	
	Acc	IIA	Acc	IIA	Acc	IIA
<i>Agreement, d = 8</i>						
1L	88.3	.78	65.2	.61	52.1	.51
2L	96.8	.96	70.3	.72	53.7	.54
4L	97.1	.96	71.5	.73	54.1	.54
6L	97.2	.96	71.8	.74	54.3	.55
<i>NPI licensing, d = 8</i>						
1L	85.1	.75	61.8	.58	50.9	.50
2L	96.4	.96	68.4	.69	51.8	.53
4L	96.7	.96	69.2	.70	52.3	.53
6L	96.8	.96	69.5	.70	52.5	.53
<i>MC-NPI, depth 3</i>						
1L	83.5	.73	62.1	.57	50.5	.50
2L	97.2	.96	69.3	.68	51.4	.52
4L	97.4	.96	70.1	.69	51.9	.52
6L	97.5	.96	70.5	.69	52.1	.53

Table 51: Attention head ablation (TF, 2 layers, 3 phenomena).

Heads	Agreement		NPI		MC-NPI	
	Acc	IIA	Acc	IIA	Acc	IIA
1	95.8	.95	95.5	.94	96.3	.95
2	96.3	.95	95.9	.95	96.8	.96
4	96.8	.96	96.4	.96	97.2	.96
8	96.9	.96	96.5	.96	97.3	.96

racy), likely by implementing redundant copies of the same operation.

I.8 Mamba State Dimension

Table 52: Mamba SSM state dimension N ablation ($d_h = 128$ except †: joint configuration $d_h = 1024$, $N = 128$; 3 seeds).

N	Agreement				NPI licensing			
	$d=0$	$d=8$	Δ	IIA	$d=0$	$d=8$	Δ	IIA
4	95.1	51.5	43.6	.51	94.8	50.5	44.3	.50
8	96.3	52.3	44.0	.52	95.7	50.9	44.8	.51
16	97.2	53.7	43.5	.54	96.8	51.8	45.0	.53
32	97.5	55.1	42.4	.56	97.1	52.8	44.3	.54
64	97.8	57.3	40.5	.58	97.3	54.1	43.2	.55
128	97.9	59.2	38.7	.60	97.4	55.5	41.9	.57
128†	98.2	64.1	34.1	.62	97.8	60.0	37.8	.59

Increasing N from 4 to 128 at $d_h = 128$ improves $d = 0$ accuracy by $< 3\%$ but $d = 8$ accuracy by only $\sim 8\%$. Within this sweep the gap Δ remains above 38% at every N . IIA increases slowly (0.51 to 0.60) but stays far below the 0.95 threshold. This is consistent with the state interference argument (Appendix B.3): even with a larger

state, each input-dependent transition still corrupts the stored signal. Even with both memory axes at their joint maxima ($d_h = 1024$, $N = 128$, row †; 3 seeds, behavioral s.d. ≤ 2.7 , IIA s.d. ≤ 0.021), the $d = 8$ gap to the Transformer remains above 30 points and IIA below 0.65: the memory budget, on either axis or jointly, is not the binding constraint.

I.9 Token Position

Table 53: DAS intervention position ablation (TF).

Position	Agreement		NPI	
	S_N	V_F	H_L	N_O
Consumed (ours)	.97	.96	.97	.96
Subject/licensor pos.	.95	.71	.94	.68
Verb/NPI pos.	.89	.93	.86	.92
Last token	.93	.94	.92	.93
Mean-pool (all)	.82	.78	.80	.77

The “consumed” position (the token at which the model must use the variable to make its prediction) yields the highest IIA for both variables. Intervening at the subject position yields high atomic IIA but low compositional IIA (the compositional variable has not yet been computed). Mean-pooling dilutes the signal across all positions. These results validate the consumed-position choice and are consistent across all phenomena.

I.10 Batch Size

Table 54: Batch size ablation (TF, Agreement).

Batch	Acc ($d=8$)	Grok step	IIA
64	96.1	18K	.95
128	96.3	15K	.96
256	96.5	14K	.96
512	96.8	12.5K	.96
1024	96.7	11K	.96

Larger batches accelerate grokking (18K $\rightarrow$ 11K steps) but do not affect the final accuracy or IIA. The phenomenon holds across all six tasks.

I.11 Model Learning Rate

Results are robust within the range 5×10^{-4} to 5×10^{-3} . The architecture ranking (TF $\gg$ LSTM $>$ Mamba) is invariant to learning rate. At 10^{-4} , grokking is delayed but still occurs for TF (accuracy 94.5%, IIA 0.93). All six phenomena show the same pattern.

Table 55: Model learning rate ablation (Agreement $d=8$).

lr	TF		LSTM		Mamba	
	Acc	IIA	Acc	IIA	Acc	IIA
10^{-4}	94.5	.93	65.8	.66	51.2	.51
5×10^{-4}	96.2	.95	69.1	.71	52.8	.53
10^{-3}	96.8	.96	70.3	.72	53.7	.54
5×10^{-3}	96.5	.95	69.5	.70	53.1	.53

I.12 SAE Feature Alignment

We train a Sparse Autoencoder (SAE) on Transformer residual-stream activations and compare the alignment of top-activated features with DAS IIA.

Table 56: SAE feature alignment vs. DAS IIA (TF, Agreement).

Method	SUBJ_NUM	VERB_FORM
DAS IIA ($r=2$)	.97	.96
SAE top-1 feat. corr.	.83	.71
SAE top-5 feat. corr.	.89	.79
SAE top-10 feat. corr.	.91	.82

SAE features capture substantial information about the causal variables (correlation up to 0.91 for SUBJ_NUM), but fall below DAS IIA because SAE features are learned for reconstruction, not for causal faithfulness. The gap is larger for compositional variables, consistent with the MIB finding that SAE features underperform DAS on causal-variable localization (Mueller et al., 2025).

I.13 Linear Probing Baseline

Table 57: Linear probe accuracy vs. DAS IIA (all 8 variables, TF).

Variable	Type	Probe acc.	DAS IIA
SUBJ_NUM	at.	.98	.97
VERB_FORM	co.	.95	.96
HAS_LIC	at.	.98	.97
NPI_OK	co.	.94	.96
ANTE_NUM	at.	.98	.97
REFL_OK	co.	.95	.96
DET_NUM	at.	.99	.98
CONC_OK	co.	.98	.97

Linear probe accuracy is comparable to or slightly above DAS IIA for atomic variables and slightly below for compositional variables. The key difference is conceptual: probing measures *decodability* (is the information present?), while DAS IIA measures *causal usage* (does the model use it as predicted?). On Mamba, probe accuracy for

SUBJ_NUM reaches 0.78 while DAS IIA is only 0.63, confirming that Mamba *encodes* subject number but does not *use* it via the prescribed algorithm.

I.14 Gradient-Based Attribution

Table 58: Attribution method comparison (TF, Agreement). Values are alignment scores (correlation with ground-truth variable importance).

Method	SUBJ_NUM	VERB_FORM
DAS (interventional)	.97	.96
Integrated Gradients	.82	.75
Attention weights	.78	.68
Input $\times$ gradient	.79	.72

Gradient-based methods consistently underperform DAS because they measure *sensitivity* (how much the output changes with input perturbation) rather than *causal mediation* (whether a specific subspace mediates the computation). Attention weights are the weakest, consistent with known limitations of attention-based interpretability (Jain and Wallace, 2019). All three gradient methods yield the same architecture ranking (TF > LSTM > Mamba) as DAS, but with smaller effect sizes, making them less suitable for the fine-grained comparisons in this paper.

J Statistical Testing

All tests use 5 random seeds unless otherwise noted. We report raw p -values alongside Bonferroni-corrected significance at $\alpha = 0.05$.

J.1 Behavioral Accuracy: TF vs. Mamba

Table 59: Paired t -test: TF vs. Mamba accuracy at max complexity (5 seeds). $\alpha_{\text{Bonf}} = 0.05/6 = 0.0083$.

Phenomenon	$\bar{\Delta}$	t	df	p	Sig.
Agr., $d=8$	43.1	44.2	4	$< 10^{-4}$	✓
NPI, $d=8$	44.6	45.8	4	$< 10^{-4}$	✓
Refl., $d=8$	44.2	45.4	4	$< 10^{-4}$	✓
Conc., $d=1$	0.7	1.8	4	0.15	✗
MC-NPI, dep. 3	45.8	47.0	4	$< 10^{-4}$	✓
CC-bind., $d=8$	45.6	46.8	4	$< 10^{-4}$	✓

All long-distance phenomena yield $p < 10^{-4}$ (significant after Bonferroni correction). Concord yields $p = 0.15$ (not significant), confirming the predicted absence of a gap.

The full ordering TF > LSTM > Mamba is statistically significant on all long-distance phenomena. Concord differences are not significant for any pair, as predicted.

Table 60: Paired t -test: TF vs. LSTM and LSTM vs. Mamba accuracy at max complexity.

Phenomenon	TF vs. LSTM		LSTM vs. Mamba	
	$\bar{\Delta}$	p	$\bar{\Delta}$	p
Agr., $d=8$	26.5	$< 10^{-4}$	16.6	$< 10^{-3}$
NPI, $d=8$	28.0	$< 10^{-4}$	16.6	$< 10^{-3}$
Refl., $d=8$	27.4	$< 10^{-4}$	16.8	$< 10^{-3}$
Conc., $d=1$	0.4	0.28	0.3	0.41
MC-NPI, dep. 3	27.9	$< 10^{-4}$	17.9	$< 10^{-3}$
CC-bind., $d=8$	29.1	$< 10^{-4}$	16.5	$< 10^{-3}$

J.2 IIA: TF vs. Mamba

Table 61: Paired t -test: TF vs. Mamba IIA (5 seeds $\times$ 3 DAS inits = 15 paired estimates per variable).

Variable	Type	$\bar{\Delta}$	t	p	Sig.
SUBJ_NUM	at.	0.34	18.9	$< 10^{-10}$	✓
VERB_FORM	co.	0.42	24.7	$< 10^{-11}$	✓
HAS_LIC	at.	0.37	17.1	$< 10^{-10}$	✓
NPI_OK	co.	0.43	22.3	$< 10^{-11}$	✓
ANTE_NUM	at.	0.35	18.5	$< 10^{-10}$	✓
REFL_OK	co.	0.42	23.1	$< 10^{-11}$	✓
DET_NUM	at.	0.28	20.3	$< 10^{-10}$	✓
CONC_OK	co.	0.32	21.8	$< 10^{-11}$	✓

All 8 variables yield $p < 10^{-10}$, significant after Bonferroni correction ($\alpha_{\text{Bonf}} = 0.05/8 = 0.00625$). The IIA gap is larger for compositional variables ($\bar{\Delta} = 0.42\text{--}0.43$) than for atomic variables ($\bar{\Delta} = 0.28\text{--}0.37$), reflecting Mamba’s partial success on the simpler retrieval step but complete failure on the Boolean combination.

Seed-level analysis. Because architectures and phenomena share the same five seeds and pipelines, we additionally report an analysis with seed as the sole unit. For each seed we average the behavioral $d = 8$ gap over phenomena; the per-seed mean Transformer–LSTM gap is at least 26 points and the Transformer–Mamba gap at least 43 points, against a between-seed s.d. below 4 points, significant under a paired t -test across the five seeds ($p < 0.001$). A mixed-effects model with architecture as a fixed effect and crossed random intercepts for seed and phenomenon confirms the architecture effect. These seed-level analyses replace any statement that pools seed-phenomenon cells as independent observations.

J.3 Atomic Before Compositional Assembly

The Wilcoxon signed-rank test yields $p = 0.031$ for each phenomenon individually (the minimum

Table 62: Assembly ordering test (TF, $\theta = 0.80$). Each phenomenon row: Wilcoxon signed-rank paired comparison of atomic vs. compositional assembly point across 5 seeds. Wins: seeds in which the atomic variable assembles before the compositional one; W : Wilcoxon signed-rank statistic ($W = 15$ is the maximum for $n = 5$); ✓: significant at $\alpha = 0.05$. The final row treats seed as the sole unit of analysis: an exact binomial test of $H_0: p = 0.5$ over $n = 5$ seeds, p -value $0.5^5 = 0.031$.

Phenomenon	Wins	W	p	Sig.
Agreement	5/5	15.0	0.031	✓
NPI licensing	5/5	15.0	0.031	✓
Reflexive	5/5	15.0	0.031	✓
Concord	5/5	15.0	0.031	✓
Seed-level	5/5	—	0.031	✓

achievable p with 5 paired samples). We do not pool the 5×4 seed-phenomenon cells into a single sign test, since the four phenomena share the same five seeds and pipelines and the cells are therefore not independent. Treating seed as the sole unit of analysis, every seed shows the ordering in all four phenomena, giving an exact binomial $p = 0.5^5 = 0.031$; a mixed-effects model on per-seed assembly-point differences with crossed random intercepts for seed and phenomenon yields a significantly positive intercept, confirming the ordering without an independence assumption.

J.4 Bootstrap Confidence Intervals

We compute 95% bootstrap CIs (1,000 replicates, percentile method) for key quantities.

Table 63: 95% bootstrap confidence intervals.

Quantity	Point est.	95% CI
TF–Mamba gap, Agr. $d=8$	43.1%	[39.2, 47.0]
TF–Mamba gap, NPI $d=8$	44.6%	[40.5, 48.7]
TF–Mamba gap, MC-NPI dep. 3	45.8%	[41.3, 50.3]
TF–Mamba gap, Conc. $d=1$	0.7%	[−0.2, 1.6]
IIA gap, SUBJ_NUM	0.34	[0.29, 0.39]
IIA gap, VERB_FORM	0.42	[0.38, 0.46]
Assembly gap (at.–co.), Agr.	2.5K	[1.5, 3.5]
Assembly gap (at.–co.), NPI	2.5K	[1.5, 3.5]

All long-distance accuracy gaps have CIs that exclude zero. The concord gap CI includes zero [−0.2, 1.6], consistent with the predicted absence of a gap. Assembly gap CIs are bounded away from zero, confirming that the atomic-before-compositional ordering is statistically robust.

J.5 Effect Sizes

Table 64: Cohen’s d for key comparisons (with 95% CI).

Comparison	Cohen’s d	Interpretation
TF vs. Mamba acc, Agr. $d=8$	12.3 [8.1, 16.5]	very large
TF vs. Mamba acc, NPI $d=8$	11.8 [7.9, 15.7]	very large
TF vs. LSTM acc, Agr. $d=8$	8.5 [5.6, 11.4]	very large
LSTM vs. Mamba acc, Agr. $d=8$	4.2 [2.7, 5.7]	very large
TF vs. Mamba IIA, S_N	7.2 [4.8, 9.6]	very large
TF vs. Mamba IIA, V_F	9.8 [6.6, 13.0]	very large
Prescribed vs. Alt-SCM IIA	5.1 [3.4, 6.8]	very large
Atomic vs. compositional assembly	3.8 [2.5, 5.1]	very large
TF vs. Mamba acc, Conc.	0.7 [−0.6, 2.0]	medium

All long-distance comparisons yield Cohen’s $d > 4$ (very large effects by conventional standards). The concord comparison yields $d = 0.7$ (medium effect), with a CI that includes zero.

J.6 Prediction Success

We test whether the 6/6 correct prediction rate is statistically significant.

Binomial test. Under H_0 : each prediction has probability $p = 0.5$ of being correct by chance. The probability of 6/6 correct is $0.5^6 = 0.016 < 0.05$.

Conservative analysis. The six predictions are not fully independent: they share the same three architectures and training pipeline. A conservative approach treats each *phenomenon* as independent but the three architectures within each phenomenon as dependent. With 6 independent phenomena and 2 possible outcomes each (gap or no gap), we have 6 binary predictions. The probability that all 6 match the predicted direction under H_0 is still $0.5^6 = 0.016$.

Permutation test. We randomly permute the architecture labels within each phenomenon and count how often the permuted assignment produces a gap pattern consistent with the succinctness prediction. Out of 10,000 random permutations, 0 produced 6/6 correct predictions, yielding $p < 10^{-4}$.

Quantitative predictions. Beyond directional predictions, we test the *ordering* of gap magnitudes. The predicted order (concord $<$ agreement $d = 0 <$ long-distance) matches the observed order exactly. A Kendall’s τ test on the 6 predicted vs. observed gap magnitudes yields $\tau = 0.93$, $p = 0.008$.

K C-Command Template Family

To test structural rather than linear operationalization, clause brackets are added to the alphabet, and four NPI template types are constructed (Table 65). *T1 (distractor)*: a licensor sits inside an embedded clause, linearly nearest to the NPI but structurally inert; every T1 item is unlicensed. *T2 (long licit)*: the structurally licit licensor appears at long linear distance. *T3 (conflict)*: both licensors are present, and only the matrix licensor determines the label. *T4 (anchor)*: the original templates, recovering 96.8/68.9/52.7 for TF/LSTM/Mamba and confirming pipeline consistency. Throughout, structural distance d_{struct} denotes the linear distance in tokens from the structurally licit licensor to the NPI; the embedded distractor, when present, is always linearly closer than the licit licensor. Training follows the main protocol of §4 over the bracket-extended alphabet, with 3 seeds per architecture; DAS follows Section D.5 under the validation/test protocol.

Table 65: C-command template family: accuracy (%), 3 seeds.

Type	TF	LSTM	Mamba
T1 distractor	92.7	41.6	24.8
T2 long licit	95.1	63.8	47.9
T3 conflict	90.9	46.7	29.6
T4 anchor	96.8	68.9	52.7

T1 is the diagnostic: since every T1 item is unlicensed, a model applying the linear-proximity heuristic answers wrongly *by system*, not by chance. Mamba’s 24.8%, far below chance, is precisely that signature; because T1 items form an all-unlicensed class, systematic application of the linear heuristic drives accuracy toward zero, and the observed value quantifies how consistently the heuristic is applied. The Transformer correctly rejects 92.7%. Causally, DAS on the structural task locates HAS_LIC at IIA = 0.944 under the validation/test protocol, and interventions on the structurally inert licensor leave predictions unchanged 96.3% of the time: the variable the Transformer computes tracks the structurally licit licensor, not the linearly nearest one. The distance scaling survives the structural operationalization: the Transformer–Mamba gap grows from 36.3 points at $d_{\text{struct}} = 2$ to 60.6 at $d_{\text{struct}} = 10$, with the Transformer nearly distance-stable (a 4.8-point drop).

L Compute Budget and Reproducibility

L.1 Per-Experiment GPU Hours

Table 66: Compute budget breakdown. Camera-ready additions comprise the pre-trained Mamba evaluation, the Pythia checkpoint trajectory, the c-command family, the λ sweep, the joint d_h/N point, and the protocol re-evaluation.

Experiment group	Configs	GPU-h
From-scratch training (TF, LSTM, Mamba)	$3 \times 6 \times 5$ seeds	45
From-scratch training (GRU, RWKV)	$2 \times 6 \times 5$ seeds	30
Width scaling	$5 \times 6 \times 2 \times 5$ seeds	150
DAS evaluation (main, 3 arch)	$3 \times 8 \times 4 \times 3$ inits	30
DAS evaluation (GRU, RWKV)	$2 \times 8 \times 4 \times 3$ inits	20
Training dynamics (check-points)	$5 \times 4 \times 15 \times 3$ inits	40
Negative controls (7 families)	~ 200 configs	25
Ablation studies (I.1–I.14)	~ 350 configs	80
Pythia DAS	$3 \times 4 \times 4 \times 3$ inits	15
$2 \times$ param baseline	$2 \times 6 \times 5$ seeds	10
Depth ablation	$5 \times 3 \times 6$ phens	20
Camera-ready additions	~ 330 configs	105
Total		~ 570

L.2 Hardware

All experiments were run on a single machine with the following specifications:

- GPU: $4 \times$ NVIDIA A100 80 GB
- CPU: $2 \times$ AMD EPYC 7763 64-Core
- RAM: 512 GB DDR4
- Storage: 2 TB NVMe SSD

Experiments were parallelized across the 4 GPUs. The experiments of the submitted version took approximately 12 days of wall-clock time; the camera-ready additions ran during the revision period. Each individual training run (from-scratch, 40K steps, $d_h = 128$) takes approximately 25 minutes on a single A100.

L.3 Software Versions

Table 67: Software versions.

Package	Version
Python	3.10.12
PyTorch	2.1.2
CUDA	12.1
pyvene	0.1.2
mamba-ssm	1.2.0
transformers (HF)	4.36.0
numpy	1.26.2
scipy	1.11.4
scikit-learn	1.3.2
matplotlib	3.8.2

The Transformer and classification head use a custom implementation (not HuggingFace transformers) for simplicity and full control over the architecture. The HuggingFace transformers library is used only for loading Pythia checkpoints. DAS is implemented via the pyvene library (Wu et al., 2024). Mamba uses the official mamba-ssm package.

L.4 Reproducibility

The full pipeline is specified in the paper: the template grammars of Section C.1 allow exact re-generation of all datasets; model configurations and training hyperparameters are given in Sections D.1 and D.3, and DAS settings in Section D.5; all random seeds are fixed and reported (Table 14). Pre-trained Pythia checkpoints are publicly available from EleutherAI (<https://github.com/EleutherAI/pythia>). We have verified that results are reproducible across two independent machines with the same software versions.

L.5 Environmental Impact

This work does not involve human subjects or personally identifiable information; all data are template-generated (Section A.1). The total compute budget is approximately 570 GPU-hours on NVIDIA A100 GPUs (Table 66). Using the Machine Learning CO₂ Impact Calculator (Lacoste et al., 2019) with North American grid estimates (0.43 kg CO₂eq/kWh) and an A100 TDP of 300 W at $\sim 70\%$ average utilization, we estimate total emissions of approximately 52 kg CO₂eq. The Pythia and Mamba models used for pre-trained validation were trained by their releasing organizations; we did not incur the cost of their pre-training.

L.6 Licenses

All third-party artifacts are used in accordance with their licenses, and our use (research evaluation of publicly released models and benchmarks) is consistent with their intended use:

- Pythia models and checkpoints: Apache License 2.0 (EleutherAI; Biderman et al., 2023)
- Official Mamba checkpoints: Apache License 2.0 (Gu and Dao, 2023)
- BLiMP benchmark data: CC-BY 4.0 (Warstadt et al., 2020)
- pyvene (DAS implementation): MIT License
- mamba-ssm package: Apache License 2.0
- PyTorch: BSD-3-Clause (Paszke et al., 2019)