Hierarchical Latent Structures in Data Generation Process Unify Mechanistic Phenomena across Scale

Anonymous authors
Paper under double-blind review

Abstract

Contemporary studies in mechanistic interpretability have uncovered many puzzling phenomena in the neural information processing of Transformer-based language models, such as induction heads, function vectors, and the Hydra effect. Some of these individual phenomena have been independently tied to different data distributional properties, while some have been loosely associated with model architecture and how Transformers process information. However, a unified understanding of the relationship between data, model architecture, and optimization remains lacking, failing to answer the fundamental question: why do these three phenomena appear universally across different model families and scales, despite their seeming disconnect? In this work, we answer this question by unifying these three phenomena as consequences of hierarchical latent structures in the data generation process, coupled with decorrelated gradients across additive model components and directional concavity in the representation geometry. We validate our theoretical results in a toy model regime and in a large-scale synthetic data regime, comparing them with language models trained on natural language data. [1]

1 Introduction

Recent investigations aiming at representative disassembly of a language model, commonly termed under the broad umbrella of mechanistic interpretability (Olah, 2022), have identified striking phenomena emerging within these models — inference-time self-repair or the *Hydra effect* McGrath et al. (2023), *induction heads* (Olsson et al., 2022), *function vectors* (Todd et al., 2024). Each of these phenomena is universal to autoregressive language models trained on large natural language corpora; yet, we lack any framework to explain why they appear or an explanation for their coincidental emergence.

We note that not all mechanistic phenomena act on the same scale/scope. For example, induction circuits are pertinent to local predictive structures of the form $[\mathtt{A}][\mathtt{B}]\cdots[\mathtt{A}] \to [\mathtt{B}]$, where the prediction of $[\mathtt{B}]$ is conditioned on matching prefix $[\mathtt{A}]$. Function vectors are generalizations of induction-like mechanisms, in which the model compresses task-specific information from in-context evidence into reusable vector representations. The Hydra effect, on the other hand, is fundamental to how predictive information is processed in the different parts of the model architecture: ablation of a model component at inference time causes another component to compensate via increased predictive influence.

Behind this seeming disconnect, however, lies some intuitive commonality. The linkage between induction heads and function vectors is fairly straightforward. Both these phenomena explain in-context learning. Recent literature (Chan et al., 2022) attribute the emergence of in-context learning to multiple data-distributional properties, such as burstiness, dynamic meaning, Zipfian distribution, etc. It is natural to assume that these properties control induction heads and function vectors as well. Chan et al. (2022) note an interesting observation: while the mentioned data-specific properties elicit in-context learning, it is exclusive to Transformers and not recurrent architectures.

Prior work posit that *copy suppression* causes Hydra effect: if model components in earlier layers predict a certain token, and this token appears earlier in the context, certain attention heads (termed copy suppressing

[1] We make our code available with the accompanying supplementary material.

heads) suppresses that copied information (McDougall et al., 2023). When the suppressed head is ablated, suppression is not in place and hence, the predictive performance does not get hurt. This observation unravels a deeper connection between Hydra effect and in-context learning: for the copy suppression to successfully elicit Hydra effect, the model needs to have similar yet redundant evidence in the context, and must be able to infer useful information from such evidences. Such ability, as we discussed, are inherently linked to induction heads and function vectors.

To this end, we seek to unify the explanations of these three unique mechanistic properties under the umbrella of data distribution process, model architecture, and learning dynamics. The common theme of all three properties is the presence of shared semantics across different parts of the input. We model this shared semantics as hierarchically structured latents in the data generation process. We first develop a theoretical framework to attribute the role of hierarchy in the elicitation of induction heads, function vectors, and the Hydra effect (Section 3). We verify the theoretical results on a strictly controlled toy model regime: Transformers trained on hierarchical data generation process exhibit the properties under consideration significantly better than a flat data generation process with same surface-level statistics (Section 4). Finally, in Section 5, we perform large-scale comparison between Transformers trained on synthetic N-gram data (a flat process), synthetic data generated by Probabilistic Context-Sensitive Grammars or PCSGs (a hierarchical process) and the OLMo-1B checkpoints (Groeneveld et al., 2024) (language model trained on natural language). Confirming the results drawn from the theoretical framework and toy model experiments, models trained on PCSG demonstrate the emergence of these three phenomena, similar to OLMo-1B, while the N-gram model fails to do so.

2 Related Work

Mechanistic phenomena in language models. Early works in mechanistic interpretability, primarily popularized by Olah (2022); Elhage et al. (2021); Olsson et al. (2022), focus on reverse engineering the computation performed by the language model as circuits. Induction circuits (Olsson et al., 2022) are the very first family of such mechanisms found in toy as well as large-scale Transformer-based language models: two-layer circuits that perform prefix-matching and copying. Subsequent work has studied how induction circuits form during training in a synthetic data regime (Singh et al., 2024). Interestingly, they point toward a emergent redundancy of induction heads, though falling short in explaining the redundancy. Chen et al. (2024) prove that gradient flow on Markov models with repeated evidence. Olsson et al. (2022)originally linked induction heads as precursor to in-context learning. Toward identifying the mechanism of few-shot in-context learning, Todd et al. (2024) demonstrate the existence of function vectors (FVs): compact, causally effective representations of in-context examples localized in a small set of attention heads. Independent contemporary work by Hendel et al. (2023) confirm the same as task vectors. Dong et al. (2026) show that task vectors naturally emerge in linear transformers trained on triplet-formatted prompts through loss landscape analysis. Across many of these discussed work, along with orthogonal research into reverse engineering circuits responsible for indirect object identification Wang et al. (2023), presence of redundant mechanisms has been pointed out. McGrath et al. (2023) present a comprehensive analysis of redundancy elicited as Hydra effect: ablating an attention layer causes downstream layers to compensate. McDougall et al. (2023) provide a partial mechanistic account of Hydra effect via copy suppression heads. Rushing & Nanda (2024) attribute Hydra effect to two different mechanisms within the model: Layernorm scaling and anti-erasure MLP neurons.

Training dynamics offer insight into emergent behaviors in neural networks, where gradual increases in scale or training progress lead to sudden qualitative changes in capability (Anderson, 1972; Wei et al., 2022). Such phase transitions have been hypothesized to arise from the rapid formation of specialized circuits (Michaud et al., 2023). Transformer architecture has been shown to possess implicit biases of gradient flow that the near-optimal model is equipped with induction heads (Chen et al., 2024). Zucchet et al. (2025) study how factual recall is acquired over training. Despite isolated understandings of different such phenomena, a unified picture remains missing. Moreover, phenomena such as Hydra effect (McGrath et al., 2023) remain unexplored from the training dynamics and data distribution lens.

Synthetic data for interpretability. Prior work has shown that model performance follows robust scaling laws determined not only by model size and compute but also by the structure of the training data (Kaplan et al., 2020; Hoffmann et al., 2022), highlighting a gap between expressibility and learnability that depends on data structure and optimization dynamics (Weiss et al., 2021; Borenstein et al., 2024). To study how data properties give rise to emergent behaviors, researchers have increasingly turned to synthetic datasets, which allow precise control over statistical and structural features while preserving key learning dynamics observed in natural language models (Elhage et al., 2022; Jain et al., 2024). However, many synthetic setups rely on flat or sequential data and fail to capture the hierarchical and recursive structure central to language. Probabilistic context-free grammars (PCFGs) offer a principled middle ground, exposing a transparent data generation process while retaining core linguistic properties, such as compositionality, ambiguity, and recursion. Previous work has used PCFGs to probe inductive biases (White & Cotterell, 2021), derive exact comparisons between learned and true distributions (Jumelet & Zuidema, 2023), and study the acquisition of hierarchical syntax (Allen-Zhu & Li, 2025). Most closely, Schulz et al. (2025) show that transformers reduce loss across grammatical components in parallel yet struggle to learn deep recursion, suggesting optimization limits rather than architectural constraints. While these studies establish PCFGs as effective tools for analyzing linguistic competence, they largely stop short of mechanistically explaining how models internalize structure.

Hu et al. (2025) demonstrate a strong precedence to our case; they coin the term *pre-pretraining* (training on formal language before natural languages) and found that hierarchical formal language data elicits linguistic generalization in natural language models.

While there have been prior effort to explain some of these mechanistic phenomena from, they remain of narrow focus: either they attribute individual phenomenon to different model components (e.g., induction heads, copy suppression heads, anti-erasure neurons, etc.) and fail to answer why they appear in the training process. Prior work that investigate the role of data distribution and training dynamics usually focus on a single phenomenon in a restrictive setting. On the contrary, we present the first theoretical and empirical unification of seemingly disconnected mechanistic phenomena as an immediate result of hierarchy in the data generation process.

3 Theoretical Results

In this section, we show that under realistic assumptions about the data generation process and model architecture, hierarchically persistent features are guaranteed to *enforce* the emergence of induction, function vectors, and the Hydra effect in an autoregressive model trained via gradient descent.

Let $X = (x_1 x_2 \cdots x_n)$ be a sequence of tokens generated by a latent hierarchical process of depth $D \geq 2$:

$$Z^{(0)} \to Z^{(1)} \to X$$

where $Z^{(0)}$ is a global latent variable, $Z^{(1)}$ is a local latent variable, and the joint distribution factorizes as:

$$p\left(X, Z^{(0)}, Z^{(1)}\right) = p\left(Z^{(0)}\right) p\left(Z^{(1)} | Z^{(0)}\right) \prod_{t=1}^{n} p\left(x_t | Z_t^{(1)}, Z^{(0)}\right)$$

We make the following assumptions about the data generation process:

Assumption 1. *Regularity.* All random variables are defined on standard Borel spaces, ensuring the existence of regular conditional distributions.

Assumption 2. *Hierarchical dependence.* There exist tokens such that

$$x_t \not\perp Z^{(0)} | Z^{(1)}, \;\; x_t \not\perp Z^{(1)} | Z^{(0)}$$

i.e., latents from different levels jointly dictate token probability.

Assumption 3. *Non-degeneracy.* Latent variables influencing prediction are statistically recoverable from the input:

$$I\left(Z^{(0)}; X_{<t}\right) > 0, \;\; I\left(Z^{(1)}; X_{<t} | Z^{(0)}\right) > 0$$

Assumption 4. *Multiple evidence streams.* There exist disjoint index sets $S_1, S_2, \cdots \subset \{1, \cdots, n\}$ such that,

$$X_{S_i} \to Z_i^{(1)} \to Z^{(0)}, \text{ for } i = 1, 2$$

and

$$Z_1^{(1)} \perp Z_2^{(1)} | Z^{(0)}$$

i.e., two disjoint evidence sets shares latent via distinct local latents.

We consider an autoregressive model:

$$p(x_t | X_{<t}) = \mathcal{M}\left(X_{<t}\right) \in \mathbb{R}^V$$

where a fixed-dimensional internal representation is mapped to logits via linear decoding W_U:

$$\mathcal{M}\left(X_{<t}\right) = W_U h^{(L)}\left(X_{<t}\right)$$

The final representation $h^{(L)}$ allows additive decomposition into K functional units f_k parametrized by θ_k:

$$h^{(L)}(X) = g\left(\sum_{k=1}^{K} f_k(X; \theta_k)\right)$$

Each f_k can be loosely described as an information processing path. A smooth, nonlinear function g then aggregates these pathways to retrieve the global latent. In practice, f_k can be either conceptualized via parallel attention heads (where g can be the MLP that aggregates), or via residual connected layers (where g is a subsequent layer that aggregates). We do not commit to either. Instead, for the theoretical argument, we rely on the following assumption:

Bayes optimal predictor for the defined data generative process can be formulated as:

$$P\left(x_t | X_{<t}\right) = \sum_{z^{(0)}, z_t^{(1)}} P\left(x_t | z^{(0)}, z_t^{(1)}\right) P\left(z_t^{(1)} | z^{(0)}, X_{<t}\right) P\left(z^{(0)} | X_{<t}\right) \tag{1}$$

Define

$$g\left(z^{(0)}, X_{<t}\right) = \mathbb{E}_{z_t^{(1)} | z^{(0)}, X_{<t}}\left[P\left(x_t | z_t^{(1)}, z^{(0)}\right)\right]$$

Then Eq. 1 can be rewritten in the following nested form:

$$P\left(x_t | X_{<t}\right) = \mathbb{E}_{z^{(0)} | X_{<t}}\left[g\left(z^{(0)}, X_{<t}\right)\right] \tag{2}$$

Theorem 1. *Under Assumptions 1-4, any finite-capacity model minimizing next-token loss must implement representation that aggregates information about $Z^{(0)}$ from multiple disjoint subsets of the context via intermediate latent structure.*

Proof: From Eq. 2, the prediction depends on inferring the global latent $P\left(z^{(0)} | X_{<t}\right)$.

By Assumption 4,

$$P\left(z^{(0)} | X_{<t}\right) = \sum_{z_i^{(1)}} P\left(z^{(0)} | z_i^{(1)}\right) P\left(z_i^{(1)} | X_{S_i}\right), \text{ for } i = 1, 2$$

and since,

$$Z_1^{(1)} \perp Z_2^{(1)} | Z^{(0)}$$

The posterior computation requires aggregation over disjoint contexts via different latent pathways. ■

Theorem 1 directly connects to the existence of induction heads and computation of function vectors. Induction mechanisms perform the context-matching and retrieval from arbitrary evidences. Function vectors are representation of the posterior estimate $P(Z^{(0)} | X_{<t})$ that are subsequently used to predict x_t.

Next, we proceed to establish the connection between Hydra effect and hierarchical data generation process.

Theorem 2. *Define the branch evidence function*

$$e_i(z^{(0)}, x_i) = \log \left(\sum_{z_i^{(1)}} p(z_i^{(1)} \mid z^{(0)}) p(x_i \mid z_i^{(1)}, z^{(0)}) \right).$$

Then the posterior distribution of the global latent satisfies

$$\log p(z^{(0)} \mid x_{1:m}) = \log p(z^{(0)}) + \sum_{i=1}^{m} e_i(z^{(0)}, x_i) - \log p(x_{1:m}).$$

Hence each conditionally independent branch contributes an additive log-evidence term toward inference of the shared latent $Z^{(0)}$.

For the complete proof of Theorem 2, see Appendix A.1. We make the following assumption about the gradients associated with these additive branch evidences implemented by the model:

Assumption 5 (Low-Interference Optimization)**.** Let

$$u_i(\theta) = \nabla_\theta \hat{e}_i(z^{(0)}, x_i)$$

denote the gradient associated with the model's estimate $\hat{e}_i$ of branch evidence e_i. Assume there exists $\varepsilon > 0$ such that

$$|\cos(u_i, u_j)| = \left| \frac{u_i^\top u_j}{\|u_i\| \|u_j\|} \right| \leq \varepsilon$$

for all $i \neq j$.

Our next theorem posits that such additive, conditionally independent branches result in approximately block-diagonal Fisher matrices.

Theorem 3. *Under the assumptions of Theorem 2 and Assumption 5, let*

$$u = \nabla_\theta \log p_\theta(z^{(0)} | x)$$

denote the score function of a posterior estimator, which can be written as

$$u = \sum_{i=1}^{m} u_i \text{ where } u_i = \nabla_\theta \hat{e}_i(z^{(0)}, x_i)$$

corresponds to the learned estimator of the branch evidence term. Define the Fisher information matrix $F = \mathbb{E}\left[uu^\top\right]$, then

$$F = \sum_{i=1}^{m} F_i + R,$$

where

$$F_i = \mathbb{E}\left[u_i u_i^\top\right]$$

and the residual coupling matrix satisfies

$$\|R\| \leq \varepsilon \sum_{i \neq j} \sqrt{\mathbb{E}[\|u_i\|^2]\, \mathbb{E}[\|u_j\|^2]}$$

Consequently,

$$F = \sum_i F_i + O(\varepsilon),$$

and therefore approaches block-diagonal form as $\varepsilon \to 0$.

For the complete proof, see Appendix A.2. An immediate result from Theorem 3 is the following corollary:

Corollary 3.1 (Emergent Specialization)**.** *Under the assumptions of Theorem 3, the optimization geometry decomposes into approximately independent Fisher subspaces associated with the evidence terms* $\{e_i\}_{i=1}^m$*. Consequently, gradient updates arising from one branch induce only* $O(\varepsilon)$ *interference on parameters primarily used to estimate another branch. As* $\varepsilon \to 0$*, optimization dynamics become increasingly decoupled, favoring the emergence of specialized computational circuits that estimate distinct branch-evidence terms.*

Now we proceed to show how specialization of implementing e_i via dedicated functional units f_{k_i} elicits Hydra effect. We make the following assumption about the representation geometry:

Assumption 6 (Directional Concavity of Readout)**.** Define the scalar readout for the target token x_t as

$$\phi(\mu) := \langle g(\mu), w_t \rangle,$$

where w_t is the row of the unembedding matrix W_U corresponding to x_t, and μ is the final representation. We assume that ϕ is twice differentiable and locally concave along directions induced by redundant latent estimators. In particular, for any representation ν and any unit contribution μ_1 corresponding to an independent estimator of $Z^{(0)}$,

$$\mathbb{E}\left[\mu_1^\top H_\phi(\nu)\, \mu_1\right] \leq 0,$$

where $H_\phi(\nu)$ is the Hessian of ϕ at ν.

Theorem 4. *Let* f_{k_1}, f_{k_2} *be functional units that specialize in estimating branched evidences* e_1, e_2 *respectively. Let*

$$\mu_i := f_{k_i}(X), \quad i = 1, 2,$$

and define the full representation

$$\mu := \sum_{k=1}^{K} f_k(X; \theta_k).$$

Let $r := \sum_{k \neq k_1, k_2} f_k(X; \theta_k)$ *denote the contribution of all other units.*

Define the predictive influence for the correct token x_t *as*

$$\Delta := \langle g(\mu_1 + \mu_2 + r), w_t \rangle,$$

and the predictive influence after ablating unit k_1 *as*

$$\tilde{\Delta}_{k_1} := \langle g(\mu_2 + r), w_t \rangle.$$

Under Assumptions 4, 5, and 6, it holds that

$$\mathbb{E}\left[\tilde{\Delta}_{k_1}\right] \geq \mathbb{E}\left[\Delta\right].$$

See Appendix A.3 for the complete proof.

4 Toy Model of Hierarchy

To complement the theoretical results of Theorems 1-4, we design a controlled experimental setup that allows us to test each claim in isolation, free from confounders present in natural-language corpora. In this section, we describe the data generation processes, model, mechanistic metrics, assumption probes, and significance-testing procedure.

4.1 Data Generation Processes

We instantiate two DGPs that share surface statistics but differ in latent structure, following the theoretical hierarchy of Section 3.

Hierarchical DGP Each input sequence of length n is divided into S fixed-length segments of length $L = n/S$. A global latent variable $Z^{(0)} \in \{1, \ldots, K_0\}$ is drawn once per sequence from a uniform prior π_0. Each segment s receives an independent local latent $Z_s^{(1)} \in \{1, \ldots, K_1\}$ drawn from the conditional table $\pi_1(Z^{(1)} \mid Z^{(0)})$, which is concentrated — each global state concentrates its mass over a distinct subset of local states, making $Z^{(1)}$ an informative but noisy signal about $Z^{(0)}$.

Within each segment, the first $L-2$ positions are *evidence tokens*, drawn from a per-$Z^{(1)}$ emission table $\mathbb{E}_{\text{ev}}[Z^{(1)}, \cdot]$ that does not depend on $Z^{(0)}$. Position $L-1$ is a deterministic [QUERY] cue token; position L is the *query token*, drawn from a joint emission table

$$\mathbb{E}_{\text{q}}[Z^{(0)}, Z^{(1)}, x] \;\propto\; \alpha_q \cdot E_0[Z^{(0)}, x] + (1 - \alpha_q) \cdot E_1[Z^{(1)}, x],$$

where $\alpha_q \in (0, 1)$ controls the relative weight of each level. Query tokens therefore depend *jointly* on both latents, instantiating Assumption 2: neither $Z^{(0)}$ alone nor $Z^{(1)}$ alone is sufficient for Bayes-optimal prediction. Optimally predicting query tokens requires aggregating cross-segment evidence (which reveals $Z^{(0)}$ through the concentrated transition π_1) with within-segment evidence (which reveals $Z^{(1)}$ directly), instantiating disjoint evidence streams of Assumption 4.

Flat DGP The flat ablation retains the same segment structure, emission vocabulary, and surface statistics, but removes the global latent entirely. Query tokens are drawn from a marginalized emission table $\mathbb{E}_{\text{q,flat}}[Z^{(1)}, x] = \sum_{z_0} \pi_0(z_0) \cdot \mathbb{E}_{\text{q}}[z_0, Z^{(1)}, x]$, so cross-segment inference is both unnecessary and impossible. The model can still exploit within-segment $Z^{(1)}$ signal, but the Bayes gap $I(Z^{(0)}; x_q \mid Z^{(1)}, X_{<t})$ is positive by construction, guaranteeing a meaningful performance differential between the two DGPs.

To rule out degenerate configurations, we reject any DGP instance where the Bayes entropy of the query token decreases by fewer than 0.08 nats across context segments (i.e. where cross-segment evidence provides negligible benefit), re-sampling the random seed until this criterion is satisfied.

We train two identical autoregressive Transformers on sequences from each DGP. Design details are described in Appendix B.1.

4.2 Metrics

We measure three mechanistic phenomena at fixed evaluation intervals during training.

Layerwise Hydra effect. For each upstream layer ℓ', we ablate its output by zeroing the residual contribution and measure the resulting change in predictive influence at every downstream layer $l > l'$. The predictive influence of layer l on sequence $X_{<t}$ is defined as $\Delta^{(l)} = \langle W_U[x_t],\, h^{(l)}(X_{<t})\rangle$, where $W_U[x_t]$ is the row of the unembedding matrix corresponding to the target token. The Hydra score for the pair (l', l) is

$$\mathcal{H}(l', l) \;=\; \mathbb{E}\Big[\Delta^{(l)}_{\text{ablate}(l')} - \Delta^{(l)}\Big],$$

and we report the scalar $\max_{l'<l} \mathcal{H}(l', l)$ as the peak compensation signal.

Head-wise Hydra effect. We repeat the same protocol at head granularity: for each attention head (i, h), we zero its value output before the output projection and measure the change in downstream predictive influence. This yields a three-dimensional tensor $\mathcal{H}[i, h, j] \in \mathbb{R}$ for all $j > i$, and a summary matrix $\mathcal{H}[i, h] = \max_{j>i} \mathcal{H}[i, h, j]$ that identifies which heads elicit the strongest downstream compensation.

Induction head score. Following Olsson et al. (2022), we construct sequences of the form $[r_0, \ldots, r_{L-1}, A, r_L, \ldots, r_{2L-2}, A]$, where A is a randomly chosen token and r_i are independent random fillers. The final token A is the query; the induction target is position L (the token immediately following the first occurrence of A). For head (l, h), the induction score is the attention weight placed at the induction target position when processing the final query:

$$\text{IS}(l, h) \;=\; \mathbb{E}_A\Big[\alpha^{(l,h)}_{2L-2,\,L}\Big],$$

where the expectation is over stimulus draws. We report $\max_{l,h} \mathrm{IS}(l,h)$, which is close to $1/T$ for a model with no induction circuits and approaches 1 for a model with a dedicated induction head. Crucially, the stimulus sequences are generated independently of the DGP test set, so the score measures a circuit property of the model rather than a statistical artifact of the training distribution.

Function vector score. We operationalize function vectors following Todd et al. (2024). For each test sequence, we designate the tokens from all but the final segment as the *source context* and the final segment alone as the *target context*. We extract the attention output of each layer at the final position of the source context, then inject it additively into the corresponding layer's residual stream during a clean forward pass over the target context. The function vector score for layer l is the mean logit increase for the correct query token induced by this intervention:

$$\mathrm{FV}(l) \;=\; \mathbb{E}[f_{\mathrm{patch},l}(X_{<t})[x_t] - f_{\mathrm{base}}(X_{<t})[x_t]]\,,$$

and we report $\max_l \mathrm{FV}(l)$.

In addition to the above-mentioned metrics, we also empirically verify the validity of Assumptions 5 and 6, since these are non-trivial assumptions about the gradient descent dynamics and representation geometry of the model.

Assumption 5 (Low-Interference Optimization). We operationalize this assumption at the granularity of individual attention heads, treating each head (l,h) as a distinct functional unit $f_{l,h}$ with residual contribution

$$f_{l,h}(X) \;=\; \big(\alpha^{(l,h)} V^{(l,h)}\big) W_O^{(h)\top} \;\in\; \mathbb{R}^D,$$

where $W_O^{(h)}$ is the column slice of the output projection corresponding to head h.

Corresponding to a forward pass in which each head contribution $f_{l,h}$ and the cross-entropy loss incurred is $\mathcal{L}$, the gradient $g_{l,h} = \partial\mathcal{L}/\partial f_{l,h}$ is available for each head. We then compute the $LH \times LH$ matrix of pairwise cosine similarities between flattened gradient vectors. The primary scalar we track is

$$\text{Gradient decorrelation} \;=\; \min_{(l,h)\neq(l',h')} \big|\cos(g_{l,h},\, g_{l',h'})\big|,$$

the *minimum* off-diagonal cosine similarity.

Assumption 6 (Directional Concavity of Readout). The readout is $\phi(v) = \langle g(v),\, w_t\rangle$, where g denotes the composition of the final layer normalisation and the linear unembedding, and v is the residual stream at the query position prior to g. For each test sequence, we collect v (the residual before the final layer norm) and u_1 (the attention output of an early layer at the same position, as a proxy for a redundant latent estimator), then compute the exact Hessian $H_\phi(v)$ via second-order automatic differentiation through the final layer norm. The directional curvature is

$$c_i \;=\; u_1^\top H_\phi(v_i)\, u_1.$$

Assumption 6 states $\mathbb{E}[c] \leq 0$; we report the mean directional curvature.

4.3 Results

Figure 1 reports the trajectory of the different metrics in models trained on hierarchical and flat DGPs. Headwise Hydra effect, induction heads, and function vectors emerge selectively under hierarchical structure. All three metrics yield consistently and substantially higher values for the hierarchical DGP across the entire training run, with near-zero counterparts in the flat DGP. The distributional separation at the end of training is unambiguous: for all three metrics across all three aggregation windows (final, peak, AUC), paired t-tests and Wilcoxon signed-rank tests reject the null at $p < 0.001$, and all comparisons survive Bonferroni correction over the full family of tests (see Figures 4 and 5 in Appendix). Effect sizes are large throughout, ranging from $d = 1.82$ for headwise Hydra (final) to $d = 4.46$ for the function vector AUC, indicating that the hierarchical condition reliably produces these phenomena across seeds rather than in isolated runs.

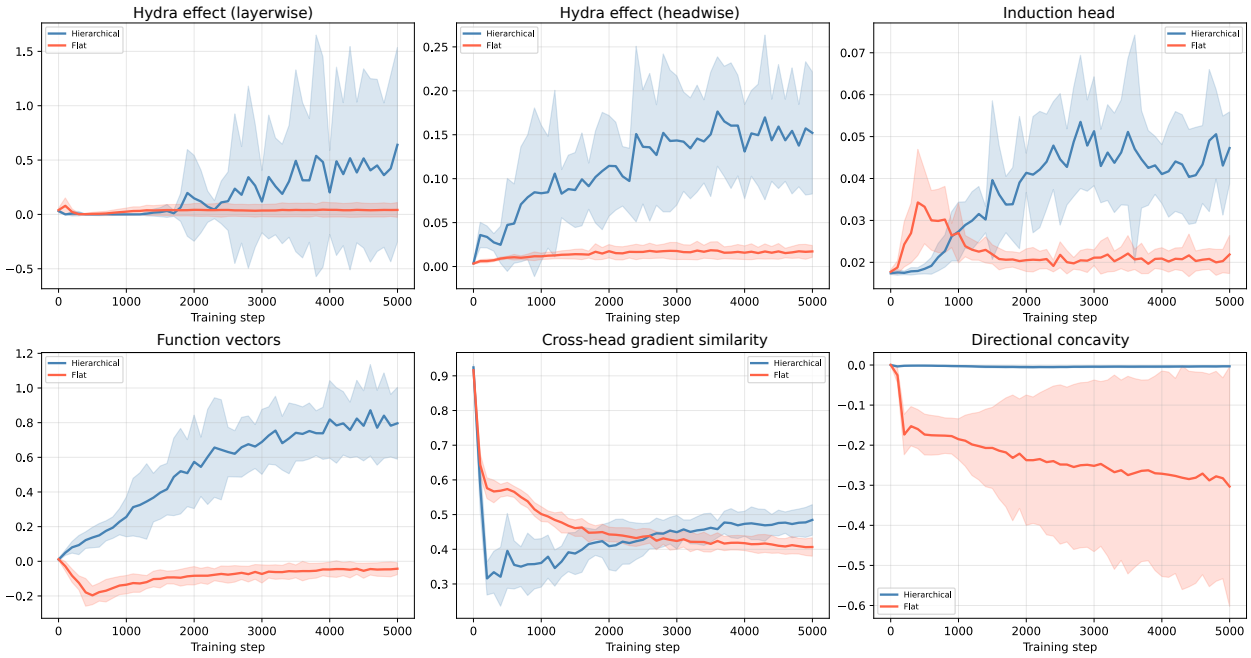

Figure 1: **Emergence of Hydra effect, induction heads, and function vectors in toy models.** All three metrics show selective emergence in the hierarchical DGP against the flat counterpart.

In case of layerwise Hydra effect, the signal is weaker (one-sided significance of $p < 0.05$ does not survive Bonferroni correction). We attribute the difference to the coarser resolution of layer-level ablation: zeroing an entire layer removes the contributions of multiple heads simultaneously, including heads that serve the same latent stream and whose ablations may partially cancel in their downstream effect. Headwise ablation isolates individual functional units and therefore provides a cleaner signal of the redundant specialisation predicted by Theorem 4. The layerwise metric remains a useful coarse indicator but is a less sensitive operationalisation of the theoretical construct than its headwise counterpart.

Probing the gradient decorrelation and directional concavity confirms the validity of Assumptions 5 and 6 on both model variants. The minimum cross-head gradient cosine similarity drops sharply in the hierarchical models at approximately the same training step at which the Hydra and induction scores rise. Note that these two properties of gradient descent and representation geometry, though necessary for Hydra effect to emerge, are not sufficient. In the flat variant, there is no sign of Hydra effect despite a more negative directional curvature and comparable gradient decorrelation. This establishes the absolute necessity of hierarchy as the X-factor to explain induction heads, function vectors, and Hydra effect.

5 Comparative Validation on Large Models

We design an experiment that isolates the effect of structure in training data while closely matching the surface statistics of natural language. We construct two data generation processes that produce text with similar token distributions, vocabulary usage, and short-range dependencies. One process contains an explicit underlying structure, while the other intentionally lacks such structure. We train identical language models on the resulting corpora using the same architecture, optimization method, and hyperparameters. At fixed intervals of x training steps, we evaluate a predefined set of model properties. We then compare these measurements to those obtained from a model trained on real-world natural language data (OLMo-1B) to validate faithfulness. The pretraining configuration can be found in Appendix D.

5.1 Data generation process

Each process draws from a fixed vocabulary of size V, denoted by the token set $\mathcal{V} = \{0, 1, \ldots, V-1\}$, where EOS is a special end-of sequence-symbol. See Appendix D for full details.

N-gram. The N-gram data generation process serves as a baseline because it represents the simplest form of generative structure, one that lacks hierarchy and recursion. It captures only local and sequential dependencies between tokens defined by a fixed history window. Sampling from a Zipf distribution preserves the statistical patterns characteristic of natural language. For an N-gram model of order n, the context h consists of the sequence of the previous $n-1$ tokens. For each unique history $h = (w_1, \ldots, w_{n-1}) \in \mathcal{V}^{n-1}$, we define a transition distribution over the next token t:

$$P(t|h) \propto \frac{1}{(t+1)^{\alpha_h}} \tag{3}$$

where the Zipf exponent α_h is sampled as $\alpha_h \sim \max\left(\mathcal{N}(\mu, \sigma^2), \alpha_{\min}\right)$ for each distinct history context. To generate a sentence $\mathbf{x} = (x_1, x_2, \ldots, x_L)$, we first determine the sentence length $L = \ell_{\min} + \epsilon, \quad$ where $\epsilon \sim \text{Zipf}(s)$. We initialize the history h_0 with a set of padding tokens. Then for $i = 1, \ldots, L$:

$$x_i \sim P(\cdot|h_{i-1})\ h_i \qquad\qquad = \text{shift}(h_{i-1}, x_i) \tag{4}$$

where the update step shift discards the oldest token in the history and appends x_i to form the new context for step $i+1$. Finally, we append the EOS token: $x_{L+1} = V - 1$.

PCSG. Similar to Charniak & Carroll (1994), we use a context-free backbone (i.e., a PCFG) and introduce context-dependent production rules at the very last stage while expanding non-terminal symbols to terminal symbols. Unlike regular grammars, which capture only local token dependencies, a PCFG introduces a hierarchical structure through nested production rules. This structure enables the generation of sentences with grammatical relationships between constituents, specifically subjects, verbs, and objects. We construct sequences using recursive non-terminal expansions that correspond to different levels (Appendix C.2).

The data generation process begins with a document symbol S, which expands into multiple document segments D_i. Each segment represents a hierarchical unit, such as a section or paragraph, that further decomposes into sentences. The `shuffle` operator applied at the document level randomizes the order of these units, preventing the model from overfitting to a fixed sequence order. Within each sentence, the production rules distinguish between two types: assertions and questions. Assertions combine subjects, verbs, and objects into simple or compound statements, while questions invert the order of these components to form interrogative patterns. This hierarchical design captures the syntactic relationships found in natural language, even though the tokens themselves remain abstract.

Terminal symbols (S_i, V_i, and O_i) correspond to vocabulary items. Each set of terminals is sampled from a Zipf distribution to reflect the frequency imbalance typical of real linguistic data. Nonterminal symbols like Subject, Verb, and Object expand into these terminal choices. Context-sensitivity is introduced via permutation functions $\text{PermuteOrder}_{D_i}$, that assigns different terminal production probabilities across different documents. Formally, for a document D_i and syntactic category C, let $\mathcal{V}_C$ be the standard ordered set of terminal symbols. The document-specific vocabulary ordering is defined by applying a random permutation $\pi_{D_i,C}$ to this set:

$$\text{PermuteOrder}_{D_i,C}(\mathcal{V}_C) = \pi_{D_i,C}(\mathcal{V}_C)$$

where $\pi_{D_i,C}$ is a permutation uniformly sampled for each specific document and category. By randomly reordering $\mathcal{V}_C$ before sampling, the stationary rank-probabilities of the Zipf distribution are mapped to different terminal symbols for each document.

As opposed to a PCFG, a PCSG retains context-dependence similar to the n-gram model, and the sole differentiator becomes hierarchy in the latent organization.

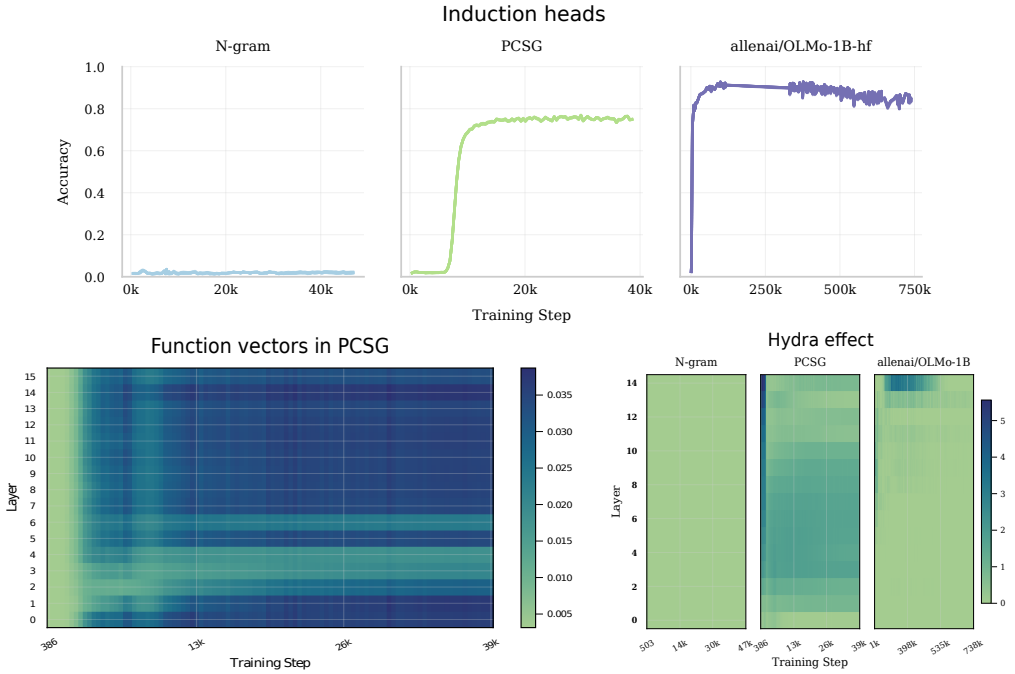

Figure 2: **Development of induction heads, function vectors and Layerwise Hydra effect in synthetic and real models.** We do not show function vectors in n-gram model since it failed to demonstrate any in-context learning, and in OLMo-1B to avoid compute expense.

5.2 Metrics

In this setup, we track the same metrics as Section 4. However, to minimize compute, we only track layerwise Hydra effect across the checkpoints. We limit the scope of ablation by focusing on the change of predictive influence of a layer upon the ablation of the immediately preceding layer.

Parse-Tree Geometry. To investigate whether the model's internal representation space encodes the hierarchical structure of the data, we employ a structural probe (Hewitt & Manning, 2019). We learn a linear transformation $B \in \mathbb{R}^{d_{\text{probe}} \times d_{\text{model}}}$ such that the squared Euclidean distance between transformed vector representations h_i, h_j approximates the distance between words x_i, x_j in the ground-truth parse tree. Formally, the probe minimizes the deviation between the tree distance $d_{\text{tree}}(x_i, x_j)$ and the probe distance $d_B(h_i, h_j)^2 = (h_i - h_j)^T (B^T B)(h_i - h_j)$. The geometric alignment is quantified using the unlabeled undirected attachment score (UUAS), which measures the percentage of edges shared between the true parse tree and the minimum spanning tree constructed from the predicted distances d_B.

5.3 Results

Figure 2 summarizes the main results for across three models and three metrics. Confirming the theory and the toy model experiments, the flat n-gram model does not develop any of the phenomena. In contrast, the PCSG shows a sharp increase in induction-related attention at approximately 6k training steps. Furthermore, its trajectory is comparable to that of the reference model, although it flattens out slightly earlier and is less unstable towards the end of training. Interestingly, improvement from function vectors substantially increases after around 6k training steps. This is also the point at which the induction heads emerge. N-gram models are not shown due to no observed existence of function vector formation.

PCSG-trained model shows substantial signs of Hydra effect. In fact, the degree of compensation is higher than OLMo-1B. Notice that in the very early stage of training, both models show compensation in the deeper layers. After training, the effect localizes in the middle layers.

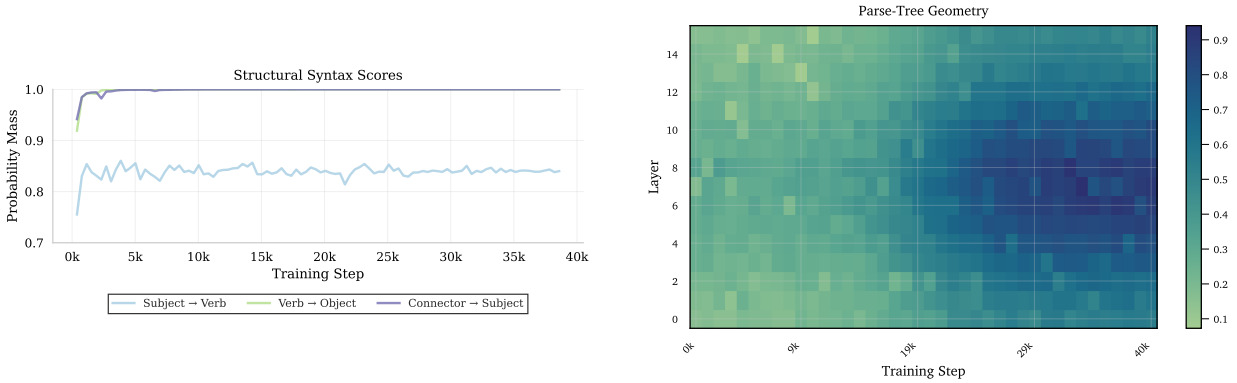

Figure 3: **Hierarchy is internalized in stages during training.** (a) Probability mass towards the next exclusively valid tokens saturates in the beginning, corresponding to shallower hierarchy learning. (b) Layer-wise structural probe accuracy rapidly improves after a substantial training (deep hierarchy learning).

To validate that the model internalizes the generative topology, we first analyze the probability mass assigned to all syntax-valid tokens (Figure 3 (a)) and find that this shallow syntax is already learned early in training, at around 4k steps. Next, to get an understanding of the more abstract syntax, we view the geometric alignment between the representation space and the ground-truth parse trees. Figure 3 (b) displays the layerwise development of the unlabeled undirected attachment score. We observe that the model progressively learns to map the Euclidean distance of its internal representations to the true tree distance of the generative grammar. This emergence is highly layer-dependent: intermediate layers (layers 5–10) exhibit the strongest structural alignment, reaching a UUAS of approximately 0.9, while earlier and later layers remain less structured. This localization suggests a dedicated syntactic processing stage within the network depth. These two observed shifts in hierarchy learning are also reflected in the loss trajectory (see Figure 6): two sharp dips in loss correspond to the emergence of shallow and deep hierarchy representations.

6 Discussion and Conclusion

In this work, we seek to unify the development of three mechanistic phenomena, namely induction heads, function vectors, and the Hydra effect, under the lens of the data generation process and optimization dynamics. We identify that hierarchical latent structure in the data generation process can be indicated as a major driver for all three phenomena. Such unification of phenomena that on the surface look very different from each other can be a stepping stone towards building a more robust understanding of language models. To facilitate a forward-moving discourse, we now highlight the overarching implications of our work.

Redundancy, interpretability, alignment. Our theoretical results establish a strong connection between hierarchical structures in the data generation process and redundant distribution of predictive power in the model. Scaling model size and complex reasoning over increasing hierarchical structures, both being the stairway to powerful model, positively reinforce computation in redundancy. This makes intervention-based interpretability (ablate a variable and record its effect) extremely challenging (Bigoulaeva et al., 2026). This extends to safety alignment as well: unless an alignment method can replace *all* possible realizations of a harmful capability, the model will remain misaligned.

Compositionality and emergence. All three phenomena investigated in large model regime demonstrate *emergent optimization* (Lubana et al., 2025; Okawa et al., 2023) — a drastic jump from a near-zero to a near-maximum value. Very often this behavior is associated with *phase shift*. Such drastic dynamics have been linked to the compositionality of the associated computation (Arora & Goyal, 2023; Lubana et al., 2025). This can be linked to learning hierarchical structures as well, both conceptually and empirically. It is natural to assume that the model's estimation of the posterior belief over shallower latents needs to be composed for the subsequent estimation of deeper latents, also evident in the internalized hierarchy in Section 5.3.

References

Zeyuan Allen-Zhu and Yuanzhi Li. Interpretability of language models for learning hierarchical structures, 2025. URL https://openreview.net/forum?id=J6qrIjTzoM.

Philip W Anderson. More is different: Broken symmetry and the nature of the hierarchical structure of science. *Science*, 177(4047):393–396, 1972.

Sanjeev Arora and Anirudh Goyal. A theory for emergence of complex skills in language models. *CoRR*, abs/2307.15936, 2023. doi: 10.48550/ARXIV.2307.15936. URL https://doi.org/10.48550/arXiv.2307.15936.

Irina Bigoulaeva, Jonas Rohweder, Subhabrata Dutta, and Iryna Gurevych. Patches of nonlinearity: Instruction vectors in large language models. *CoRR*, abs/2602.07930, 2026. doi: 10.48550/ARXIV.2602.07930. URL https://doi.org/10.48550/arXiv.2602.07930.

Nadav Borenstein, Anej Svete, Robin Chan, Josef Valvoda, Franz Nowak, Isabelle Augenstein, Eleanor Chodroff, and Ryan Cotterell. What languages are easy to language-model? A perspective from learning probabilistic regular languages. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 15115–15134. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.807. URL https://doi.org/10.18653/v1/2024.acl-long.807.

Stephanie C. Y. Chan, Adam Santoro, Andrew K. Lampinen, Jane X. Wang, Aaditya K. Singh, Pierre H. Richemond, James L. McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/77c6ccacfd9962e2307fc64680fc5ace-Abstract-Conference.html.

Eugene Charniak and Glenn Carroll. Context-sensitive statistics for improved grammatical language models. In Barbara Hayes-Roth and Richard E. Korf (eds.), *Proceedings of the 12th National Conference on Artificial Intelligence, Seattle, WA, USA, July 31 - August 4, 1994, Volume 1*, pp. 728–733. AAAI Press / The MIT Press, 1994. URL http://www.aaai.org/Library/AAAI/1994/aaai94-111.php.

Siyu Chen, Heejune Sheen, Tianhao Wang, and Zhuoran Yang. Unveiling induction heads: Provable training dynamics and feature learning in transformers. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/7aae9e3ec211249e05bd07271a6b1441-Abstract-Conference.html.

Yuxin Dong, Jiachen Jiang, Zhihui Zhu, and Xia Ning. Understanding task vectors in in-context learning: Emergence, functionality, and limitations. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=CLBVilFk7N.

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

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy Models of Superposition, September 2022.

Jaden Fiotto-Kaufman, Alexander R Loftus, Eric Todd, Jannik Brinkmann, Caden Juang, Koyena Pal, Can Rager, Aaron Mueller, Samuel Marks, Arnab Sen Sharma, Francesca Lucchetti, Michael Ripa, Adam Belfki, Nikhil Prakash, Sumeet Multani, Carla Brodley, Arjun Guha, Jonathan Bell, Byron Wallace, and David Bau. Nnsight and ndif: Democratizing access to foundation model internals. 2024. URL https://arxiv.org/abs/2407.14561.

Dirk Groeneveld, Iz Beltagy, Evan Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Raghavi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, Will Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah A. Smith, and Hannaneh Hajishirzi. Olmo: Accelerating the science of language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 15789–15809. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.841. URL https://doi.org/10.18653/v1/2024.acl-long.841.

Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9318–9333, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.624. URL https://aclanthology.org/2023.findings-emnlp.624/.

John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1419. URL https://aclanthology.org/N19-1419/.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. *CoRR*, abs/2203.15556, 2022. doi: 10.48550/ARXIV.2203.15556. URL https://doi.org/10.48550/arXiv.2203.15556.

Michael Y. Hu, Jackson Petty, Chuan Shi, William Merrill, and Tal Linzen. Between circuits and Chomsky: Pre-pretraining on formal languages imparts linguistic biases. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9691–9709, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.478. URL https://aclanthology.org/2025.acl-long.478/.

Samyak Jain, Robert Kirk, Ekdeep Singh Lubana, Robert P. Dick, Hidenori Tanaka, Tim Rocktäschel, Edward Grefenstette, and David Scott Krueger. Mechanistically analyzing the effects of fine-tuning on procedurally defined tasks. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=A0HKeKl4Nl.

Jaap Jumelet and Willem H. Zuidema. Transparency at the source: Evaluating and interpreting language models with access to the true distribution. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, Findings of ACL, pp. 4354–4369. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-EMNLP.288. URL https://doi.org/10.18653/v1/2023.findings-emnlp.288.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *CoRR*, abs/2001.08361, 2020. URL https://arxiv.org/abs/2001.08361.

Ekdeep Singh Lubana, Kyogo Kawaguchi, Robert P. Dick, and Hidenori Tanaka. A percolation model of emergence: Analyzing transformers trained on a formal language. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=0pLCDJVVRD.

Callum McDougall, Arthur Conmy, Cody Rushing, Thomas McGrath, and Neel Nanda. Copy Suppression: Comprehensively Understanding an Attention Head. *CoRR*, abs/2310.04625, 2023. doi: 10.48550/ARXIV.2310.04625. URL https://doi.org/10.48550/arXiv.2310.04625.

Thomas McGrath, Matthew Rahtz, János Kramár, Vladimir Mikulik, and Shane Legg. The hydra effect: Emergent self-repair in language model computations. *CoRR*, abs/2307.15771, 2023. doi: 10.48550/ARXIV.2307.15771. URL https://doi.org/10.48550/arXiv.2307.15771.

Eric J. Michaud, Ziming Liu, Uzay Girit, and Max Tegmark. The quantization model of neural scaling. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/5b6346a05a537d4cdb2f50323452a9fe-Abstract-Conference.html.

Maya Okawa, Ekdeep S Lubana, Robert Dick, and Hidenori Tanaka. Compositional abilities emerge multiplicatively: Exploring diffusion models on a synthetic task. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 50173–50195. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/9d0f188c7947eacb0c07f709576824f6-Paper-Conference.pdf.

Chris Olah. Mechanistic interpretability, variables, and the importance of interpretable bases. *Transformer Circuits Thread*, 2(4), 2022.

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

Cody Rushing and Neel Nanda. Explorations of self-repair in language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pp. 42836–42855. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/rushing24a.html.

Laura Ying Schulz, Daniel Mitropolsky, and Tomaso A. Poggio. Unraveling syntax: How language models learn context-free grammars. *CoRR*, abs/2510.02524, 2025. doi: 10.48550/ARXIV.2510.02524. URL https://doi.org/10.48550/arXiv.2510.02524.

Aaditya K. Singh, Ted Moskovitz, Felix Hill, Stephanie C. Y. Chan, and Andrew M. Saxe. What needs to go right for an induction head? A mechanistic study of in-context learning circuits and their formation. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pp. 45637–45662. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/singh24c.html.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=AwyxtyMwaG.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023. doi: 10.48550/ARXIV.2307.09288. URL https://doi.org/10.48550/arXiv.2307.09288.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=NpsVSN6o4ul.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022, 2022. URL https://openreview.net/forum?id=yzkSU5zdwD.

Gail Weiss, Yoav Goldberg, and Eran Yahav. Thinking like transformers. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, Proceedings of Machine Learning Research, pp. 11080–11090. PMLR, 2021. URL http://proceedings.mlr.press/v139/weiss21a.html.

Jennifer C. White and Ryan Cotterell. Examining the inductive bias of neural language models with artificial languages. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pp. 454–463. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.ACL-LONG.38. URL https://doi.org/10.18653/v1/2021.acl-long.38.

Nicolas Zucchet, Jorg Bornschein, Stephanie C.Y. Chan, Andrew Kyle Lampinen, Razvan Pascanu, and Soham De. How do language models learn facts? dynamics, curricula and hallucinations. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=vBcGnragkr.

A Complete proofs

A.1 Proof of Theorem 2

Proof. Starting from the assumed factorization,

$$p(z^{(0)}, z^{(1)}_{1:m}, x_{1:m}) = p(z^{(0)}) \prod_{i=1}^{m} p(z^{(1)}_i \mid z^{(0)}) p(x_i \mid z^{(1)}_i, z^{(0)}).$$

Conditioning on $z^{(0)}$ and marginalizing the local latents gives

$$p(x_{1:m} \mid z^{(0)}) = \sum_{z^{(1)}_{1:m}} \prod_{i=1}^{m} p(z^{(1)}_i \mid z^{(0)}) p(x_i \mid z^{(1)}_i, z^{(0)}).$$

Because the summand factorizes over i, the sum separates:

$$p(x_{1:m} \mid z^{(0)}) = \prod_{i=1}^{m} \left(\sum_{z_i^{(1)}} p(z_i^{(1)} \mid z^{(0)}) p(x_i \mid z_i^{(1)}, z^{(0)}) \right).$$

Applying Bayes' rule,

$$p(z^{(0)} \mid x_{1:m}) = \frac{p(z^{(0)}) p(x_{1:m} \mid z^{(0)})}{p(x_{1:m})}$$

Substituting the previous expression yields

$$p(z^{(0)} \mid x_{1:m}) = \frac{p(z^{(0)}) \prod_{i=1}^{m} \left(\sum_{z_i^{(1)}} p(z_i^{(1)} \mid z^{(0)}) p(x_i \mid z_i^{(1)}, z^{(0)}) \right)}{p(x_{1:m})}$$

Taking logarithms gives

$$\log p(z^{(0)} \mid x_{1:m}) = \log p(z^{(0)}) + \sum_{i=1}^{m} \log \left(\sum_{z_i^{(1)}} p(z_i^{(1)} \mid z^{(0)}) p(x_i \mid z_i^{(1)}, z^{(0)}) \right) - \log p(x_{1:m}).$$

Recognizing the definition of e_i,

$$e_i(z^{(0)}, x_i) = \log \left(\sum_{z_i^{(1)}} p(z_i^{(1)} \mid z^{(0)}) p(x_i \mid z_i^{(1)}, z^{(0)}) \right),$$

we obtain

$$\log p(z^{(0)} \mid x_{1:m}) = \log p(z^{(0)}) + \sum_{i=1}^{m} e_i(z^{(0)}, x_i) - \log p(x_{1:m}),$$

which proves the result. ■

A.2 Proof of Theorem 3

Proof. Expanding the score decomposition gives

$$u = \sum_i u_i.$$

Therefore

$$F = \mathbb{E} \left[\left(\sum_i u_i \right) \left(\sum_j u_j \right)^\top \right].$$

Expanding the product,

$$F = \sum_i \mathbb{E}[u_i u_i^\top] + \sum_{i \neq j} \mathbb{E}[u_i u_j^\top].$$

Define

$$F_i = \mathbb{E}[u_i u_i^\top]$$

and

$$R = \sum_{i \neq j} \mathbb{E}[u_i u_j^\top].$$

It remains to bound R.

By Assumption 5,

$$|u_i^\top u_j| \leq \varepsilon \|u_i\| \|u_j\|.$$

Taking expectations,

$$\left|\mathbb{E}[u_i^\top u_j]\right| \leq \varepsilon \mathbb{E}\left[\|u_i\| \|u_j\|\right].$$

Applying Cauchy–Schwarz,

$$\mathbb{E}\left[\|u_i\| \|u_j\|\right] \leq \sqrt{\mathbb{E}[\|u_i\|^2]\, \mathbb{E}[\|u_j\|^2]}.$$

Therefore

$$\left|\mathbb{E}[u_i^\top u_j]\right| \leq \varepsilon \sqrt{\mathbb{E}[\|u_i\|^2]\, \mathbb{E}[\|u_j\|^2]}.$$

Summing over all off-diagonal pairs yields

$$\|R\| \leq \varepsilon \sum_{i \neq j} \sqrt{\mathbb{E}[\|u_i\|^2]\, \mathbb{E}[\|u_j\|^2]}.$$

Hence

$$F = \sum_i F_i + O(\varepsilon).$$

Taking the limit $\varepsilon \to 0$ gives

$$F \to \sum_i F_i,$$

which is block diagonal across evidence streams.

■

A.3 Proof of Theorem 4

Proof. Define the scalar function $\phi(\mu) := \langle g(\mu), w_t \rangle$. Let

$$\nu := \mu_2 + r,$$

so that

$$\Delta = \phi(\nu + \mu_1), \quad \tilde{\Delta}_{k_1} = \phi(\nu).$$

By Corollary 3.1 and Assumption 4, the representations μ_1 and μ_2 are functions of disjoint evidence streams X_{S_1} and X_{S_2}, respectively, and are conditionally independent given the global latent $Z^{(0)}$. Moreover, both μ_1 and μ_2 serve as estimators of the same latent variable $Z^{(0)}$.

Conditioned on $Z^{(0)}$ and ν, the contribution μ_1 represents an independent estimate of the same underlying signal already partially captured by ν. Therefore, its residual contribution satisfies

$$\mathbb{E}[\mu_1 \mid Z^{(0)}, \nu] \approx 0.$$

Using a second-order Taylor expansion of ϕ around ν, we obtain

$$\phi(\nu + \mu_1) = \phi(\nu) + \nabla\phi(\nu)^\top \mu_1 + \frac{1}{2}\mu_1^\top H_\phi(\nu)\, \mu_1 + R_3,$$

where R_3 denotes higher-order terms.

Taking conditional expectation given $Z^{(0)}$ and v, the first-order term vanishes:

$$\mathbb{E}[\nabla\phi(\nu)^\top \mu_1 \mid Z^{(0)}, \nu] = \nabla\phi(\nu)^\top \mathbb{E}[\mu_1 \mid Z^{(0)}, \nu] \approx 0$$

Thus,

$$\mathbb{E}[\phi(\nu + \mu_1) \mid Z^{(0)}, \nu] \approx \phi(\nu) + \frac{1}{2}\mathbb{E}\left[\mu_1^\top H_\phi(\nu)\, \mu_1 \mid Z^{(0)}, \nu\right].$$

By Assumption 6, the second-order term is non-positive:

$$\mathbb{E}\left[\mu_1^\top H_\phi(\nu)\, \mu_1 \mid Z^{(0)}, \nu\right] \leq 0$$

Hence,

$$\mathbb{E}[\phi(\nu + \mu_1) \mid Z^{(0)}, \nu] \leq \phi(\nu)$$

Taking expectation over $Z^{(0)}$ and ν, we obtain

$$\mathbb{E}[\phi(\nu + \mu_1)] \leq \mathbb{E}[\phi(\nu)],$$

i.e.,

$$\mathbb{E}[\Delta] \leq \mathbb{E}[\tilde{\Delta}_{k_1}].$$

This completes the proof. ■

B Toy Model Details

B.1 DGP and Model Configuration

The model uses Rotary Position Embeddings (RoPE) and pre-layer normalization, with L_{model} layers, H attention heads per layer, and model dimension D. All hyperparameters — architecture, optimizer, learning rate schedule, and training budget — are held fixed across both DGP conditions. The loss mask excludes [QUERY] cue positions so that the training signal derives entirely from evidence and query tokens.

B.2 Statistical Testing

We assess whether each mechanistic effect and assumption probe is selectively and significantly stronger for models trained on the hierarchical DGP than for those trained on the flat DGP.

Experimental protocol We train N_{seeds} independent pairs of models, one per DGP, sharing the same random seed within each pair but differing in DGP type. Each pair constitutes a matched observation, eliminating seed-level variance from the comparison. At E evenly spaced evaluation checkpoints during training, we record all metrics described in Section 4.

Aggregation For each metric time series $\{m_e\}_{e=1}^{E}$, we compute three scalar summaries: (i) the *final* value at the last checkpoint, (ii) the peak value $\max_e m_e$ over training, and (iii) the area under the curve (AUC) $\frac{1}{E_{\text{final}}-E_1}\int m(e)\,\mathrm{d}e$ (approximated via the trapezoidal rule and normalized by the training span). Testing all three windows guards against timing differences in the emergence of each phenomenon.

For each metric, the hypothesis is: H_1: hierarchical $>$ flat.

Statistical tests For each metric–window combination we apply two one-sided paired tests across the N_{seeds} matched pairs: a paired t-test (parametric) and the Wilcoxon signed-rank test (non-parametric). We use one-sided tests because the direction of each effect is dictated by theory; two-sided p-values are also reported for completeness. Effect sizes are quantified by Cohen's d in the paired formulation, $d = \bar{\delta}/s_\delta$, where $\bar{\delta}$ is the mean within-pair difference and s_δ its standard deviation. Multiple-comparison correction is applied via Bonferroni over all metric–window tests simultaneously; we report both corrected and uncorrected significance decisions.

C Generative Processes Details

C.1 N-gram

Table 2 shows the parameterization for the N-gram data generation process.

C.2 PCSG

The PCSG production rules are defined in Figure 7 and parameterized as per Table 3.

D Model & Training Specifications

This appendix details the specifications of our custom model based on the LLaMA 2 architecture (Touvron et al., 2023). It follows the same architectural design but uses adjusted configuration values derived from an informed, proportional downscaling relative to `allenai/OLMo-1B-hf` to maintain a comparable performance trajectory. We used four H100 GPUs; the total amount required for training was one day. Analysis was done using the `nnsight` library (Fiotto-Kaufman et al., 2024).

D.1 Loss Trajectories

Figure 6 shows that training loss for the model based on the N-gram data generation, the PCFG data generation, and the reference model all follow a power-law trajectory. Interestingly, the PCFG shows a sharp decrease after an initial flat line. Overall, the Figure shows that the loss trajectories across all three settings are comparable, thereby establishing a baseline for subsequent analyses. Without this quantitative alignment, any proceeding comparison, such as generalization behavior, would be limited, as significant deviations in loss dynamics would indicate that the models had not reached comparable representational regimes. Thus, these results serve as a sanity check confirming that all models exhibit similar convergence behavior.

Table 1: Complete hyperparameter specification for the toy experiments. *Hierarchical only* marks parameters that apply exclusively to HierarchicalDGP; all other parameters are shared between both data generation processes unless otherwise noted.

Parameter	**Value**	**Parameter**	**Value**
Data generation (shared)		*Data generation (hierarchical only)*	
Vocabulary size V	64	Global latent states K_0	4
Sequence length n	128	Local latent states K_1	16
Number of segments S	16	π_1 concentration	30.0
Segment length $L = n/S$	8	Embedding noise σ	0.1
Embedding scale	4.0	Query mixing weight α_q	0.8
Evidence tokens per segment	6 ($= L - 2$)	Min. Bayes gap ΔH	0.08 nats
Model architecture		*Optimiser*	
Model dimension D	64	Optimiser	AdamW
Number of layers	6	Learning rate	3×10^{-4}
Attention heads H	8	$(\beta_1,\ \beta_2)$	(0.90, 0.95)
Head dimension D/H	8	Weight decay	0.1
FFN hidden ratio	4×	Gradient clipping	1.0
Position encoding	RoPE	LR warmup steps	500
Max. context length	512	LR schedule	Constant
Training		*Evaluation & significance testing*	
Training steps	5,000	Test set size	512
Batch size	64	Evaluation interval	200 steps
Sequence sampling	Online (w/ replacement)	Independent seeds N	10
Model vocab size	65 ($V + 1$)	Gradient samples	256
		A6 Hessian samples	32
		Statistical tests	Paired t, Wilcoxon
		Correction	Bonferroni
		Hypothesis direction	One-sided

Parameters	
Sentences	400M
Zipf exponent	$\mathcal{N}(2.0, 1.2)$, min 1.2
Length exponent	2.0
Sequence length	10–1010

Table 2: N-gram data generation configuration.

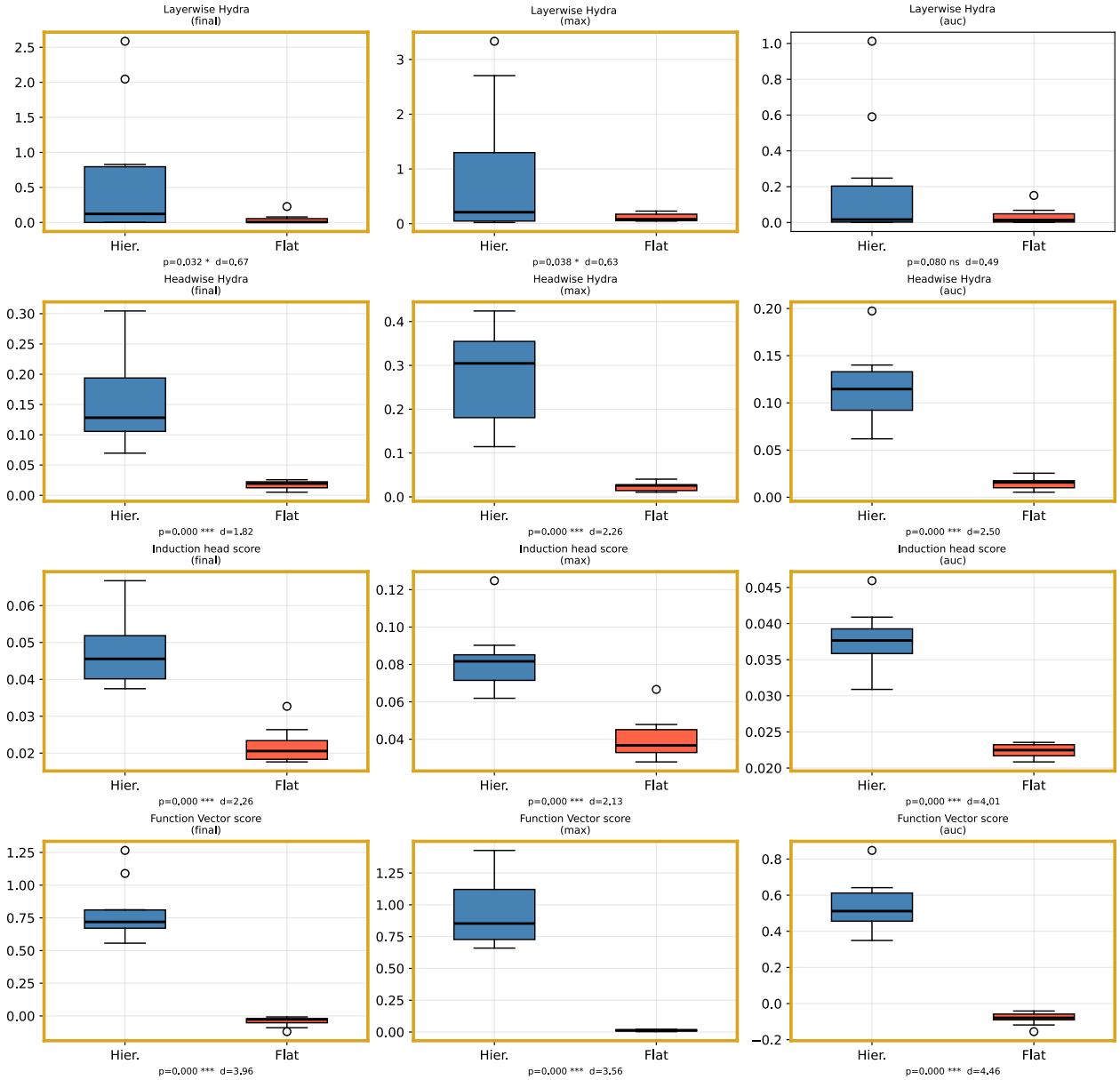

Figure 4: **Distribution of different metrics tracked.** Gold bordered subplots are statistically significant ($p < 0.05$ with one-sided t-test).

Parameters	
Documents	6.5M
Document Repetitions	10
Sections per Document	10
Paragraphs per Section	20
Zipf Exponent	1.0
Subject Percentage	0.3
Object Percentage	0.3
Verb Percentage	0.3
Connector Percentage	0.1

Table 3: PCSG data generation configuration.

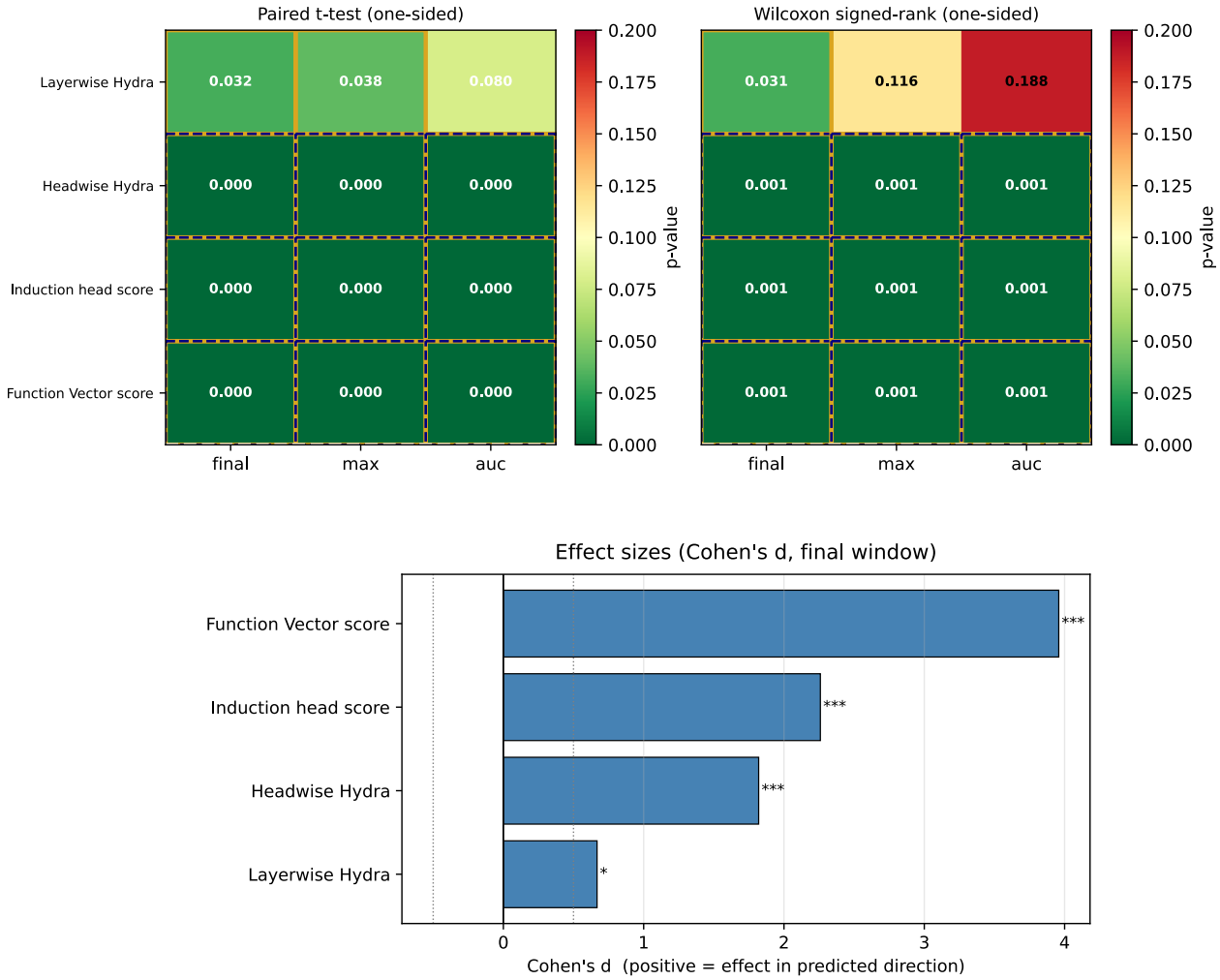

Figure 5: **Significance test results and effect sizes across four metrics.** Gold bordered subplots are statistically significant ($p < 0.05$ with one-sided t-test). Navy-dashed subplots survive significance after Bonferroni correction.

Architecture	
Parameters	17M
Dimension	256
Layers	16
Attention Heads	4
Vocabulary Size	1000
Context Length	512
Training Details	
Training Tokens	10B
Learning Rate	3×10^{-4}
Batch Size	1024
Warmup Steps	2000
Weight Decay	0.1
Gradient Clipping	1.0

Table 4: LLaMA-based 17M model configuration.

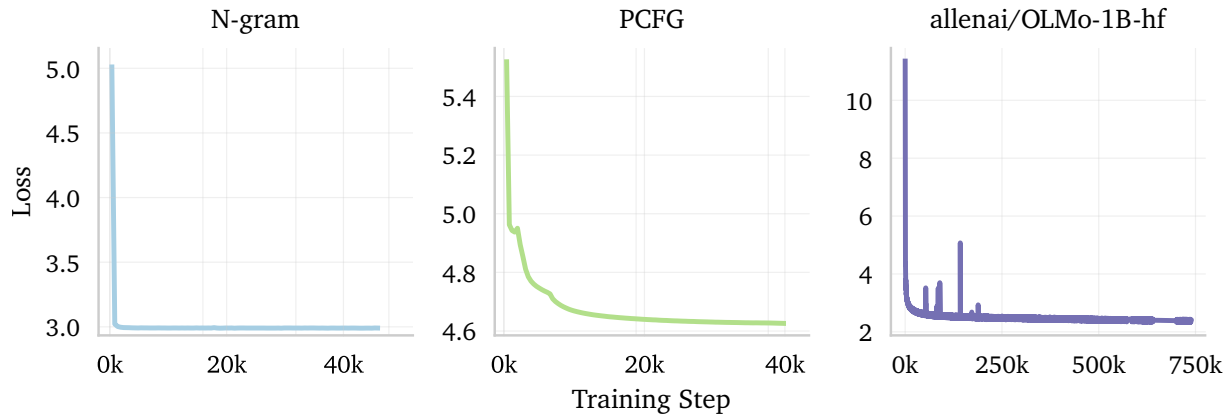

Figure 6: **Training loss across training.** The models exhibit power-law loss decay throughout training. The N-gram model shows an initial plateau followed by two subsequent drops, whereas the other two models display a steady power-law decline.

$$
\begin{array}{lcl}
\mathcal{S} & ::= & \text{shuffle}\big(\mathcal{D}_1^A \ \cdots \ \mathcal{D}_P^A\big) \\
D_i & ::= & \mathit{Section}^B \\
\mathit{Section} & ::= & \mathit{Paragraph}^C \\
\mathit{Paragraph} & ::= & \mathit{Sentence}^D \\
\mathit{Sentence} & ::= & \mathit{SentenceType}\ \text{EOS} \\
\mathit{SentenceType} & ::= & \mathit{Assertion} \ \mid \ \mathit{Question} \\
\mathit{Assertion} & ::= & \mathit{Subject\ Verb\ Object} \ \mid \ \mathit{Assertion\ Connector\ Assertion} \\
\mathit{Question} & ::= & \mathit{Verb\ Subject\ Object} \\
\mathit{Subject}(D_i) & ::= & \text{PermuteOrder}_{D_i}(\mathit{Subject}) \\
\mathit{Subject} & ::= & \text{Zipf}(\text{S}_1 \cdots \text{S}_\text{E}) \longrightarrow \text{S}_1 \ \mid \ \cdots \ \mid \ \text{S}_\text{E} \\
\mathit{Verb}(D_i) & ::= & \text{PermuteOrder}_{D_i}(\mathit{Verb}) \\
\mathit{Verb} & ::= & \text{Zipf}(\text{V}_1 \cdots \text{V}_\text{F}) \longrightarrow \text{V}_1 \ \mid \ \cdots \ \mid \ \text{V}_\text{F} \\
\mathit{Object}(D_i) & ::= & \text{PermuteOrder}_{D_i}(\mathit{Object}) \\
\mathit{Object} & ::= & \text{Zipf}(\text{O}_1 \cdots \text{O}_\text{G}) \longrightarrow \text{O}_1 \ \mid \ \cdots \ \mid \ \text{O}_\text{G} \\
\mathit{Connector} & ::= & \text{C}_1 \ \mid \ \cdots \ \mid \ \text{C}_\text{H}
\end{array}
$$

$\text{shuffle}(X_1, \ldots, X_P)$

$\pi \leftarrow \{1, \ldots, P\}$
return $X_{\pi(1)}, \ldots, X_{\pi(P)}$

$\text{PermuteOrder}_{D_i}(X_1, \ldots, X_n)$

$\sigma \leftarrow$ permutation rule for D_i
return $X_{\sigma(1)}, \ldots, X_{\sigma(n)}$

$\text{Zipf}(T_1, \ldots, T_E, \alpha)$

Sample $T_k \sim p_k$
return T_k

Figure 7: Production rules of the PCSG.