On the Emergence and Test-Time Use of Structural Information in Large Language Models

Michelle Chao Chen[1,2*], Moritz Miller[1,2], Bernhard Schölkopf[1,2], Siyuan Guo[2,3]
[1]ETH Zurich, [2]Max Planck Institute for Intelligent Systems, [3]University of Cambridge

Abstract

Learning structural information from observational data is central to producing new knowledge outside the training corpus. This holds for mechanistic understanding in scientific discovery as well as flexible test-time compositional generation. We thus study how language models learn abstract structures and utilize the learnt structural information at test-time. To ensure a controlled setup, we design a natural language dataset based on linguistic structural transformations. We empirically show that the emergence of learning structural information correlates with complex reasoning tasks, and that the ability to perform test-time compositional generation remains limited.

1 Introduction

Nature consists of structure and scale, building a hierarchical level of abstract structures to enable responsive and adaptive systems. Guo and Schölkopf (2025) postulates that learning, too, follows the laws of physics under the principle of least action. This work examines how structural information emerges under current learning paradigms and evaluates models' compositional generalization at test time. The goal is to provide evidence about their ability to generate genuinely new knowledge that is not present in the training corpus in a controlled setting.

With the successful application of large models on language, to study structure, we use linguistic structure as a synthetic playground to understand how LLMs learn and compose structure to enable adaptable prediction. One interesting aspect of language comprehension is the ability to compose existing building blocks to comprehend unseen new combinations. This paper studies the problem *how* machines comprehend structural information and *whether* they can utilize learnt knowledge for test-time composition.

To ensure a controlled synthetic playground, we generate a natural language dataset based on *Transformational Grammar (TG)* (Chomsky, 1957; Radford, 1988). This allows us to analyze *whether* and *how* the model learns the emergence of structure during training, analyze *whether* they can compose learnt structures at test-time, and provide evidence on where in the model this behavior occurs. By doing so, we shine light on how LLMs can generate sentences beyond those directly observed in the corpus. Our contributions are:

- We introduce a natural language dataset based on linguistic structural transformations to formally study structural information in language (Section 3.1).
- We show empirical results that (1) the emergence of structural information during learning correlates with the success of complex reasoning tasks (Section 4.1), and (2) the test-time compositional generation is still limited under fine-tuning (Section 4.2).
- We perform ablation studies to identify which network components are responsible for learning structural information (Section 4.3).

2 Related Work

Compositional Generalization. Research from cognitive sciences and linguistics argues that humans generalize understanding in language to unseen concepts by interpreting known components, i.e., primitives, and reorganizing basic building blocks to comprehend unseen combinations (Chomsky, 1957; Montague, 1970, 1974; Fodor and Pylyshyn, 1988). Such generalization ability is also advantageous for machines to have in terms of prediction robustness under

*Correspondence: `michelle.chao.chen@mailbox.org`

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1449–1465

Prompt for NP-Raising Transformation

Instruction: Generate a sentence with an NP-raising transformation structure. Create both an input sentence (with expletive *"it"* and embedded clause) and its corresponding output sentence (with raised subject).

Transformation pattern: `[It][raising verb][that[NP VP]] → [NP][raising verb][to VP]`

Examples:
```
Input: It seems that John is happy.
Output: John seems to be happy.
Input: It appears that the weather will improve.
Output: The weather appears to improve.
Input: It happens that she knows French.
Output: She happens to know French.
```

Instructions: Create one pair of sentences following this NP-raising pattern. Use raising verbs such as *seem, appear, happen, prove,* or *turn out*. Use the provided words naturally in your sentences. Provide both the input sentence (with expletive "it") and the transformed output sentence (with raised subject) using the format below.

Words: `[WORDS]`

Response format:
```
Input: [your input sentence here]
Output: [your output sentence here]
```

Model Output

```
Input: It appears that the button is broken.
Output: The button appears to be broken.
```

Figure 1: Example of a model prompt for the NP-raising transformation task (top) and a sample output generated by the model (bottom).

out-of-distribution generalization and, more importantly, generation of new knowledge not existed in the training corpora. Lake and Baroni (2018), for example, introduces the SCAN dataset on which models are asked to translate natural language commands into primitive-based action protocols. Despite near-perfect in-distribution performance, it is difficult for GRUs (Chung et al., 2014), LSTMs (Hochreiter and Schmidhuber, 1997) and Transformers (Vaswani et al., 2017) to perform structural generalization to unseen concepts (Kim and Linzen, 2020). Putting this into perspective, Ontanon et al. (2022) observes improving structural generalization capabilities by varying the model design and Wold et al. (2025) quantifies the difficulty of systematic generalization by introducing an entropy-based measure.

Mechanistic Interpretability. Controlled linguistic generation in language has also been used to probe and understand the inner workings of Transformers. Allen-Zhu and Li (2023), for example, studies how Transformers learn hierarchical language structure through synthetic generation of CFGs. The authors discover an implicit dynamic programming algorithm in the model that enables correct next-token prediction. Others probe the residual stream for grammatical number (Lasri et al., 2022; Ferrando and Costa-jussà, 2024) or to address the binding problem (Feng and Steinhardt, 2024). Relevant to our methodological setup, McCoy et al. (2020) studies the sequence-to-sequence model's (Sutskever et al., 2014) ability to transform a declarative statement into a question through the Question Formation Task. This setup is akin to Transformational Grammar (Chomsky, 1957; Radford, 1988) which operates over derived structures instead of hierarchical configurations as CFGs do.

Causality. A probabilistic formal treatment of structure and its advantages is manifested in causal studies (Peters et al., 2017; Pearl, 2009). Learning structural knowledge from either observational or interventional data enables downstream tasks such as causal effect estimation (Guo et al., 2024b; Robertson et al., 2025), counterfactual reasoning

(Miller et al., 2025), robust prediction under out-of-distribution shifts (Guo et al., 2024a; Perry et al., 2022), and representation and world model understanding (Reizinger et al., 2025a; Lei et al., 2022; Reizinger et al., 2025b). It has long been established that structural knowledge is impossible to discover from observational data alone (Pearl, 2009); recent work (Guo* et al., 2023) show that heterogeneous observational data is capable of recovering unique causal structures. We thus hypothesize that the model learning from a diverse training corpus also learns implicit re-usable linguistic structures that would be useful for (1) mechanistic understanding; (2) robust composition for new knowledge generation.

3 Problem Motivation

Transformational grammar posits that sentences have two levels of representation: a deep structure, which captures the underlying semantic and syntactic relations between elements of a sentence, and a surface structure, which represents the realized or spoken form (Radford, 1988). Transformational rules operate on the deep structure to derive the surface structure, thereby explaining how speakers can generate and comprehend an unlimited number of grammatical sentences.

We adopt this perspective to examine the mechanisms that allow language models to exhibit similar generative capabilities. Our dataset contains input sentences paralleling deep structures and output sentences produced through custom transformation rules designed to achieve analogous effects.

3.1 Dataset

To allow a rigorous study of how models learn, represent, and apply linguistic transformations, we design a dataset based on Transformational Grammar.

Definition. *Let $s \in \mathcal{S}$ denote a natural language sentence for $\mathcal{S}$ the space of all sentences. Let $T \in \mathbb{N}$ define the number of possible transformations on the input sentence s, and by $R \geq T$ the number of mutually disjoint sets on which at least one of the T transformations can be performed. We partition $\mathcal{S}$ into $R+1$ disjoint sets and define the sequence of disjoint sets $(\Omega_r)_{r \in \{1,\dots,R+1\}}$. For $r \in \{1, \dots, R\}$, we finally define transformations $F_r : \Omega_r \to \mathcal{S}$.*

Proposition. *In our setup, any transformation on sentence $s \in \mathcal{S}$ satisfies either of the two cases:*

1. *for $r \in \{1, \dots, R\}$, $F_r : \Omega_r \to \Omega_{r'}$ such that $F_r(s) \in \Omega_{r'}$ for $r' \in \{1, \dots, R\}$,*
2. *for $r \in \{1, \dots, R\}$, $F_r : \Omega_r \to \Omega_{R+1}$ such that $F_r(s) \in \Omega_{R+1}$.*

In our dataset, each F_r corresponds to a specific transformation, such as passivization, NP-raising, or question formation. For example, a sentence $s \in \Omega_r$ like "The scientist discovered the formula" may be transformed by F_{passive} into "The formula was discovered by the scientist," which lies in another subset $\Omega_{r'}$. Since multiple transformations can be applied sequentially (e.g., applying NP-raising after passivization), the output of one transformation can serve as the input to another. However, certain transformations eventually reach an *absorbing state* Ω_{R+1}—a set of sentences on which no further valid transformations can be applied under our defined grammar (e.g., once a sentence is already in passive voice, applying passivization again has no effect).

Dataset Features. A summary of the dataset is provided in Table 4.

- **Single-level transformations.** A transformation operator $\mathcal{A}$ is applied to a base sentence s, producing $\mathcal{A}(s)$.
- **Nested transformations.** Multiple operators are applied sequentially, e.g., $\mathcal{B}(\mathcal{A}(s))$, where the output of one transformation becomes the input to the next. For instance, starting from the base sentence "The artist applied the makeup for the photoshoot," applying passivization yields "The makeup was applied for the photoshoot by the artist," and subsequently applying question formation produces "Was the makeup applied for the photoshoot by the artist?".
- **Compositional ambiguity.** Not all transformations can be meaningfully composed: applying I-Movement (question formation) after Extraposition can yield ungrammatical outputs (e.g., transforming "The book disappeared on the table" into "Did the book disappeared on the table?" violates tense agreement). Thus, models must implicitly identify compatible patterns before applying multiple transformations.

In total, we define ten distinct transformation types grouped into five broader syntactic categories, such as movement, raising, and passivization. Our dataset includes several types of transformation structures (see Table 1).

Dataset Construction. For sentence generation, we use DeepSeek-V3 (DeepSeek-AI et al., 2025) with instructions. To improve consistency in generation, we prompt the model to produce input–output pairs together rather than generating base sentences first and then applying transformations separately. Prompts include high-quality examples, explicit descriptions of the transformation to be applied, and a set of required words sampled from the TinyStories (Eldan and Li, 2023) vocabulary to increase lexical variety. An example of such prompts is shown in Figure 1. We generate roughly 2000 samples for each single-level transformation and 500 samples for each nested transformation and perform filtering for duplicates and low-quality examples containing repetitions or irrelevant tokens. We performed the experiments with an extended dataset and motivate our transformations in section A.4. Our code is available at `https://github.com/MichelleChaoChen/data-generation`.

4 Experiments

We use our dataset to investigate three questions:

- The emergence of structural understanding during learning (see Section 4.1)
- The ability to compose structures in generation (see Section 4.2)
- Ablation studies on model component contribution (see Section 4.3)

4.1 Emergence of Structures

The first group of experiments investigates *whether* and *when* models learn grammatical structures in their sentence representations.

Methodology. A sentence with n tokens is represented as $s = (t_1, t_2, \ldots, t_n)$. Residual stream activations at layer ℓ denotes as $h_i^{(\ell)} \in \mathbb{R}^d$. From the last layer L, we obtain a sentence-level representation by mean pooling across the token-level $\bar{h}(s) = \frac{1}{n}\sum_{i=1}^{n} h_i^{(L)}$. For a transformation $\mathcal{A}(\cdot)$ applied to base sentence s, we compute a difference vector $\Delta_A(s) = \bar{h}(A(s)) - \bar{h}(s)$, and an ℓ_2 distance between the base and transformed sentence to capture the structural information $d(s, A(s)) = \|\bar{h}(A(s)) - \bar{h}(s)\|_2$.

Experiments. *(1) Visualization.* We cluster the $\Delta_A(s)$ vectors across transformations using K-means clustering with $k = 10$ transformation types and apply dimensionality reduction via PCA and t-SNE (van der Maaten and Hinton, 2008) for visualization. We compute cosine similarities between different vectors and analyze separability scores to quantify how distinctly different transformation types cluster in representation space. *(2) Structure emergence.* Using d as a metric for structural understanding in latent representations, we analyze its evolution across Pythia-410M (Biderman et al., 2023) checkpoints sampled at approximately exponentially spaced training steps throughout training. To assess the relationship between syntactic representation development and language modeling capability, we evaluate each checkpoint on two complementary metrics: (1) perplexity on pure language modelling tasks, such as WikiText-103 (Gu et al., 2024) and Paloma benchmarks (Magnusson et al., 2024), and (2) accuracy on standardized benchmarks from the LM Evaluation Harness (Gao et al., 2024) which focus more on natural language reasoning and knowledge.

Results. *(1) Development of Internal Representations.* We observe that different sentence transformations form distinct and separable clusters in the embedding space (see Figure 8). This pattern appears consistently across both PCA projections and t-SNE visualizations as seen in Figure 7, indicating that the model captures shared structural regularities. Furthermore, embeddings of transformations in the same category (e.g., NP Passive) form clusters close to each other, suggesting that the model consistently encodes these structural variations.

(2) Emergence of structure. Most notably in Figure 2, the average L2 norm between base and transformed sentence representations increases steadily throughout training, reflecting that the model becomes more sensitive to syntactic alterations. A sharp rise occurs around step 64k, which likely marks a phase transition where structural distinctions become more pronounced.

(3) Correlation with General Language Modeling Performance. To examine how representational changes relate to general modeling ability, we compare these findings to standard performance metrics. Figure 2 (bottom) shows that perplexity on language modeling tasks decreases sharply during early training steps. On the LM Harness benchmark, the Pythia-410M model shows limited improvement on causal or physical reason-

Table 1: Overview of the major syntactic transformations represented in our dataset. Each transformation type is paired with a short linguistic description and an illustrative example showing a base sentence s and its transformed version $\mathcal{T}(s)$.

Transformation	Description	Example
Extraposition	Moves prepositional phrases from within noun phrases to sentence-final position	**Base** s: The book on the table disappeared. **Output** $\mathcal{T}(s)$: The book disappeared on the table.
I-Movement	Moves auxiliaries or modals to sentence-initial position to form questions	**Base** s: She can swim. **Output** $\mathcal{T}(s)$: Can she swim?
NP Passive	Converts active sentences to passive with clear subject–object alternation	**Base** s: The scientist discovered the formula. **Output** $\mathcal{T}(s)$: The formula was discovered by the scientist.
NP Raising	Converts expletive or embedded clauses to raised-subject structures	**Base** s: It seems that John is honest. **Output** $\mathcal{T}(s)$: John seems to be honest.
V-Movement	Integrates separated infinitive or auxiliary components into a single clause	**Base** s: The children; to play outside. **Output** $\mathcal{T}(s)$: The children play outside.

ing tasks—expected given its smaller size—but the L2 norm trends align more closely with gains in core language modeling performance, such as next-word prediction. This suggests that the refinement of structural representations coincides with the model's development of linguistic competence.

Overall Insight. These results indicate that sentence representations exhibit a sudden phase transition as training progresses. The perplexity loss does not correlate closely with the structural information learned, but the structural information correlates with the more complex reasoning tasks, indicating that reasoning may require structural information.

4.2 Structural Compositional Generalization

To evaluate a trained model's ability to use the internal representation of the transformations, we perform both full parameter finetuning and parameter-efficient finetuning using LoRA adapters (Hu et al., 2022) with LLaMA3-8B (Grattafiori et al., 2024) (rank = 16).

In particular, given a base sentence s and transformation types $\mathcal{A}$ and $\mathcal{B}$, we train the model to produce the correct transformed output $\mathcal{T}(s)$ from prompts of the form shown in Figure 3. For nested transformations, the prompt specifies the combined transformation label (e.g., `Transform` ($\mathcal{A}$+$\mathcal{B}$)). During fine-tuning, both intermediate (i.e. application of one transformation) and final outputs are provided. At inference, only the initial input and the transformation name are given, and the model must generate both the intermediate and final transformed sentences.

Evaluation Protocol. We assess performance using exact match accuracy and partial match accuracy (word overlap using Jaccard similarity ≥ 0.8). For nested transformations producing multiple outputs, we check whether the prediction contains all expected results. The evaluation covers both *in-distribution* transformation types and crucially tests *compositional generalization* to unseen nested transformations (e.g., training on $\mathcal{A}(\mathcal{B}(s))$ and $\mathcal{B}(\mathcal{C}(s))$, then testing on $\mathcal{A}(\mathcal{C}(s))$).

Disentanglement Hypothesis. We specifically investigate whether models can perform nested transformations *without observing intermediate results* during training. This is based on the theoretical evidence that structure is discoverable from observational data (Guo* et al., 2023) and the hypothesis that next-token prediction performs implicit disentanglement automatically (Zhang et al., 2024). Thus we finetune an additional model where the intermediate results of nested transformations are not shown. In particular, a model trained only on a diverse set of input-output pairs $(s, \mathcal{A}(\mathcal{B}(s)))$, $(s, \mathcal{B}(\mathcal{C}(s)))$ can successfully generate $\mathcal{A}(\mathcal{C}(s))$ for novel combinations. This suggests the model learns to disentangle and compose the individual transformation operations $\mathcal{A}$ and $\mathcal{C}$ at test-time, rather than merely memorizing specific transformation sequences. We believe this is one of the desirable features of next-generation AI capability and we ask the question whether the learned representations

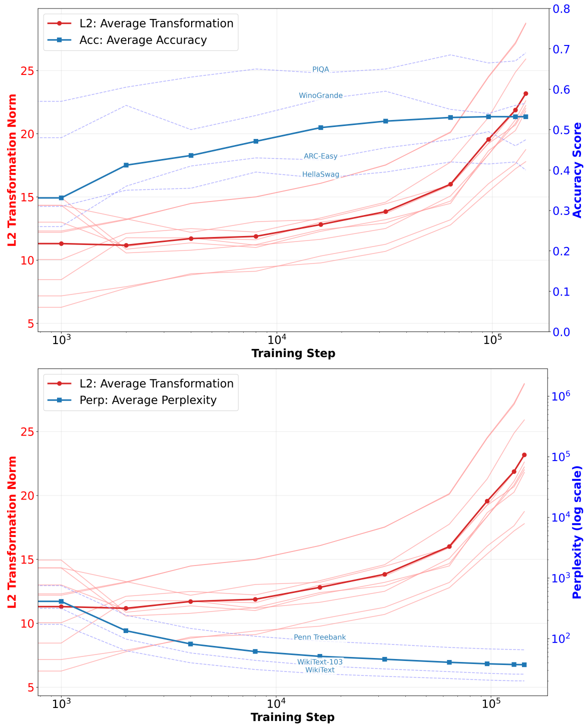


Figure 2: L2 transformation norms and Pythia-410M performance training. Red lines show syntactic transformation embedding differences. Blue lines show model performance: (Top) downstream task accuracy on reasoning tasks. (Bottom) language modeling perplexity on WikiText and Paloma datasets. Individual metrics shown as dashed lines, averages as solid lines. For error bars, see Figure 9

$\Delta A(s)$ and $\Delta C(s)$ support true compositional generation at test-time.

Results. *(1) Full-Parameter Fine-tuning.* Full-parameter fine-tuning performed poorly, likely due to the limited dataset size leading to unstable optimization and overfitting. The model failed to generalize to transformations seen during training, thus we focused subsequent experiments, including OOD evaluations, on LoRA-based fine-tuning.

(2) LoRA Fine-tuning. Using Low-Rank Adaptation (LoRA), we fine-tuned 0.52% of the parameters with significant improvement from full parameter fine-tuning as shown in Table 2. On the validation set, the model achieves high partial-match accuracy but rarely produces exact matches for nested transformations when the intermediate step (i.e., the result of the first transformation) is not shown. This suggests that while the model captured aspects of the transformation rules, it does not fully disentangle the transformation operations, which is further supported by the evaluation results on the unseen transformations (OOD).

Fine-tuning Prompt Example

During fine-tuning, both the transformation instruction and the target output are provided to the model. The model learns to predict the transformed output.

```
Transform (A): The teacher graded the
exams.
Output: The exams were graded by the
teacher.
```

Evaluation Prompt

At inference, only the transformation instruction and input sentence are provided. The model must generate the corresponding output.

```
Transform (A): The teacher graded the
exams.
Output:
```

Figure 3: Prompt formats used during fine-tuning and inference.

(3) LoRA Fine-tuning with Truncated Dataset. To further probe the ability to perform structural composition, we check the effect of providing the first intermediate result (e.g., $\mathcal{B}(s)$) for novel transformation pairs. Performance improved in this setting, indicating that the model benefits from intermediate structural guidance during multi-step transformations.

In the variant of the dataset, the intermediate results for nested transformations are removed, making them appear as single-step transformations distinguishable only by the transformation name at the beginning of the prompt. The LoRA model trained on this dataset achieves comparable performance on both validation and OOD sets when intermediate results are available during training.

Overall Insight. These findings suggest that the model fine-tuned in a small data regime cannot flexibly re-compose out-of-distribution to produce new knowledge. Nested transformations can, to some extent, be treated as extended single-step transformations. The model's performance shown in row 3 and 5 on Table 2 supports the idea that providing explicit intermediate steps during training and inference aids generalization across compositional structures to a small extent but not decisively. Further details of our experiments are found in section A.5. Our code is available at https://github.com/MichelleChaoChen/finetune-structural-sentences.

Table 2: **Comparison of fine-tuning methods across transformation settings.** We report exact and partial match accuracy for three model variants: (1) regular full-parameter fine-tuning, (2) LoRA fine-tuning, and (3) LoRA fine-tuning on sequences without intermediate results. Results highlight the generalization gap across transformation complexity and out-of-distribution (OOD) settings.

Dataset / Model Variant	**Full parameter**		**LoRA**		**LoRA (No-Intermediate)**	
	Exact	**Partial**	**Exact**	**Partial**	**Exact**	**Partial**
Single Transformations	54.80%	60.10%	96.40%	98.80%	96.40%	98.60%
Double Transformations	37.20%	40.80%	0.00%	86.95%	0.00%	81.89%
Double w/ Intermediate	–	–	82.74%	84.56%	74.37%	75.79%
OOD (A+C, H+E combinations)	–	–	0.00%	45.83%	0.00%	49.25%
OOD w/ Intermediate (A+C, H+E)	–	–	4.92%	9.85%	8.54%	32.16%

4.3 Ablation studies on attention heads and MLPs

The final set of experiments employs causal intervention analysis to identify which network components are most responsible for syntactic transformations. We perform systematic ablation studies by zeroing the outputs of individual attention heads and MLP blocks, then measuring the resulting changes in next-token probability distributions (Nanda and Bloom, 2022). The final set of experiments employs causal intervention analysis to identify which network components are most responsible for syntactic transformations. We perform systematic ablation studies by zeroing the outputs of individual attention heads and MLP blocks, then measuring the resulting changes in next-token probability distributions (Nanda and Bloom, 2022).

Formally, let $\mathbf{s} = (t_1, \ldots, t_n)$ denote an input sequence, and let t^* be the target token representing the correct syntactic transformation. For any component c (either attention head (l, h) or MLP block l), we define:

$$p_{\text{clean}}(t^* \mid \mathbf{s}) = \text{softmax}(\mathbf{W}_U \mathbf{h}_n^{(L)})_{t^*} \quad (1)$$

$$p_{\text{ablated}}^{(c)}(t^* \mid \mathbf{s}) = \text{softmax}(\mathbf{W}_U \tilde{\mathbf{h}}_n^{(L)})_{t^*} \quad (2)$$

where $\mathbf{h}_n^{(L)}$ is the final hidden state, $\tilde{\mathbf{h}}_n^{(L)}$ is the hidden state under intervention on component c, and $\mathbf{W}_U$ is the unembedding matrix. The causal contribution of component c is then:

$$\Delta p^{(c)} = p_{\text{clean}}(t^* \mid \mathbf{s}) - p_{\text{ablated}}^{(c)}(t^* \mid \mathbf{s})$$

To trace the evolution of transformation-relevant information through the network, we apply layer-wise decoding by computing intermediate probability distributions:

$$p^{(l)}(t^* \mid \mathbf{x}) = \text{softmax}(\mathbf{W}_U \text{LayerNorm}(\mathbf{h}_n^{(l)}))_{t^*}$$

for each layer $l \in \{0, 1, \ldots, L\}$, where $\mathbf{h}_n^{(0)}$ represents the embedding layer output and $\mathbf{h}_n^{(l)}$ is the residual stream after layer l.

By analyzing the distribution of $\Delta p^{(c)}$ values across all components and the trajectory of $p^{(l)}(t^* \mid \mathbf{x})$ across layers, we intend to identify the specific attention heads and MLP blocks most critical for each syntactic transformation, as well as the layers where transformation-relevant computations primarily occur.

In addition, we construct linear probes to identify directions in the residual stream that differentiate between transformation types. For each transformation type F_n and layer i, we use Linear Discriminant Analysis (LDA) to find an optimal separating direction. Specifically, we treat this as a binary classification problem: transformation F_n versus all other transformations $F_{m \neq n}$.

Let $\mathbf{r}_{F_n,i}^{(j)}$ denote the last-token residual activation at layer i for the j-th example of transformation type F_n, and let $\mathbf{r}_{\neg F_n,i}^{(k)}$ denote the last-token residual activation at layer i for the k-th example of any other transformation type. The LDA probe direction $\mathbf{v}_{F_n,i}$ is computed as:

$$\mathbf{v}_{F_n,i} = \mathbf{S}_W^{-1}(\boldsymbol{\mu}_{F_n,i} - \boldsymbol{\mu}_{\neg F_n,i}) \quad (3)$$

where $\boldsymbol{\mu}_{F_n,i} = \frac{1}{N_{F_n}} \sum_{j=1}^{N_{F_n}} \mathbf{r}_{F_n,i}^{(j)}$ and $\boldsymbol{\mu}_{\neg F_n,i} = \frac{1}{N_{\neg F_n}} \sum_{k=1}^{N_{\neg F_n}} \mathbf{r}_{\neg F_n,i}^{(k)}$ are the mean residual activations for transformation T_n and all other transformations, respectively, and $\mathbf{S}_W$ is the pooled within-class covariance matrix. We train these probes on the training set and evaluate their discriminative power on the test set.

Results. We restrict our analysis to single-level transformations to avoid confounding effects from multiple simultaneous syntactic operations, allowing us to clearly measure the contributions of atten-

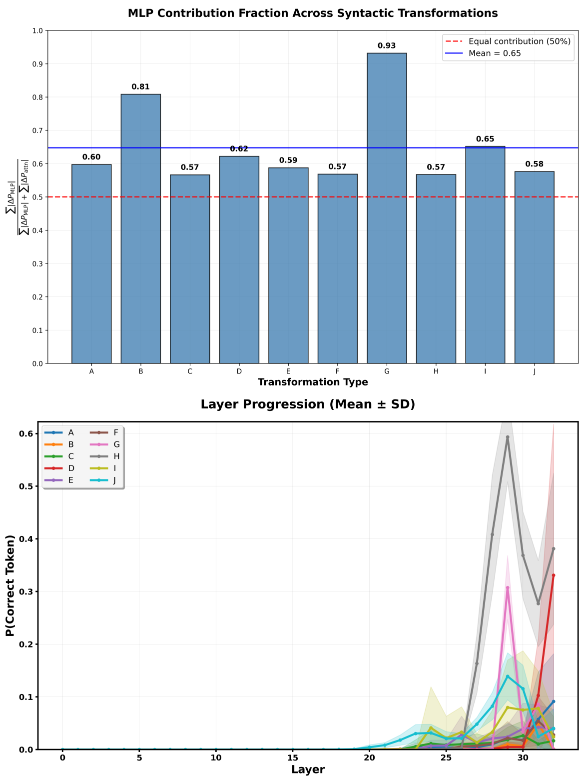


Figure 4: (Top) Relative contribution of MLP compared to multi-head attention (Bottom) Progression of probability of predicting the correct token for each transformation over all layers

tion heads and MLP layers to individual transformation types.

(1) Late-Layer Concentration. Our analysis reveals that for any given transformation type, multiple attention heads contribute meaningfully to predicting the correct target token, with no single head acting as a bottleneck for the transformation process. However, the probability of predicting the token indicative of the transformation increases sharply in the final third of the network (layers 24-32), where 38% of the contribution is concentrated. As shown in Figure 4 (bottom), the probability of predicting the correct token is near zero prior to layer 20.

(2) Component-Level Preferences. Despite the distributed nature of processing, clear systematic biases emerge at the component level. MLP blocks consistently contribute more to syntactic transformation success than attention mechanisms, accounting for 65% of total causal contribution versus 35% for attention. In Figure 4, the proportion of total contribution coming from the MLP is computed by summing over every layer. This measure relative changes in predictive probability under ablation us-

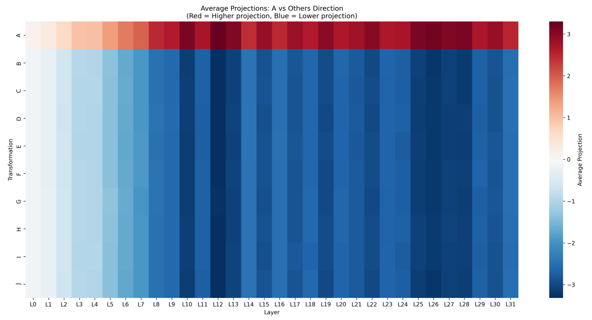


Figure 5: Heatmap showing average projections of intermediate representations onto the transformation A vs. others direction across all 32 layers. Each row represents a different transformation type (A-J), with transformation A contrasted against all other transformations. Red regions indicate higher projection values (representations more similar to transformation A), while blue regions indicate lower projection values (representations more dissimilar to transformation A).

ing a fixed unembedding matrix for consistency. The lower signal in earlier layers partly reflect limited decodability rather than absence of structural information.

(3) Representation Quality. Complementing our causal analysis, linear discriminant analysis of intermediate representations shows near-perfect separation between transformation types, indicating that the model forms distinct internal representations for different syntactic operations, consistent with Section 4.1. This can additionally be understood as evidence for the linear representation hypothesis (Park et al., 2024). In Figure 5, we show the projection values of the separating vector between transformation $\mathcal{A}$ and all other transformations. The average projection monotonically increases throughout the layers.

5 Conclusion

We discussed the emergence of learning syntactic transformations during fine-tuning. We observe that LLMs achieve near-perfect accuracy on single transformations, and 74.37% to 82.74% exact accuracy on double transformations when providing intermediate representations and LoRA fine-tuning on Llama3-8B. Our findings resonate with recent literature as both full parameter updates as well as OOD evaluation on double transformations substantially decrease performance (Kim and Linzen, 2020). To do syntactic transformations, models rely on the MLP sublayer and distinguish between different transformations at the final layers. We also provide evidence that the separation between concepts emerges as finetuning progresses and that this separation is linearly encoded in the model's residual stream. We finally introduce a natural language dataset on a broad variety of linguistic structural transformations and thereby show that LLMs can learn structural information during fine-tuning.

Limitations

Our analysis focuses only on English transformations, limiting generalizability to the cross-linguistic patterns predicted by transformational grammar theory. Languages differ in their canonical word order, structural rules, and in which syntactic operations are valid/required. For example, in English, I-movement turns "She can swim" into "Can she swim?" through auxiliary fronting 5. In contrast, Chinese forms a yes–no question as "She can swim?" without auxiliary movement, which reflects a different structural process. The single-level transformation approach, while enabling clean causal attribution, may conflate multiple sub-operations within each syntactic transformation that could be attributed to distinct network components if decomposed further. Additionally, our findings on an 8B model may not generalize to larger scales, though scaling would require more complex datasets to avoid memorization. Future work could address these through cross-linguistic studies, finer-grained operation decomposition, and evaluation on larger models with appropriately challenging transformation tasks.

Acknowledgments

M.C.C and M.M acknowledges Florent Draye for his insights and guidance on mechanistic interpretability and Hsiao-Ru Pan for feedback on the first draft. S.G. acknowledges the constructive discussions with Andreas Opedal and Ryan Cotterall on the manuscript, particularly regarding insights into computational linguistics.

References

Zeyuan Allen-Zhu and Yuanzhi Li. 2023. Physics of Language Models: Part 1, Learning Hierarchical Language Structures. *SSRN Electronic Journal*. Full version available at https://ssrn.com/abstract=5250639.

Stella Biderman, Hailey Schoelkopf, Quentin 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. *Preprint*, arXiv:2304.01373.

Noam Chomsky. 1957. *Syntactic Structures*. Mouton and Co., The Hague.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Y. Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. 2023. Faith and fate: Limits of transformers on compositionality. *Preprint*, arXiv:2305.18654.

Ronen Eldan and Yuanzhi Li. 2023. Tinystories: How small can language models be and still speak coherent english? *Preprint*, arXiv:2305.07759.

Jiahai Feng and Jacob Steinhardt. 2024. How do language models bind entities in context? In *The Twelfth International Conference on Learning Representations*.

Javier Ferrando and Marta R. Costa-jussà. 2024. On the similarity of circuits across languages: a case study on the subject-verb agreement task. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10115–10125, Miami, Florida, USA. Association for Computational Linguistics.

Jerry A. Fodor and Zenon W. Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. The language model evaluation harness.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Albert Gu, Karan Goel, and Christopher R´e. 2024. Wikitext-103 dataset.

Siyuan Guo and Bernhard Schölkopf. 2025. Physics of learning: A lagrangian perspective to different learning paradigms. *arXiv preprint arXiv:2509.21049*.

Siyuan Guo*, Viktor Tóth*, Bernhard Schölkopf, and Ferenc Huszár. 2023. Causal de finetti: On the identification of invariant causal structure in exchangeable data. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, volume 36, pages 36463–36475. Curran Associates, Inc. *equal contribution.

Siyuan Guo, Jonas Wildberger, and Bernhard Schölkopf. 2024a. Out-of-variable generalization for discriminative models. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Siyuan Guo, Chi Zhang, Karthika Mohan, Ferenc Huszár*, and Bernhard Schölkopf*. 2024b. Do finetti: On causal effects for exchangeable data. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, volume 37, pages 127317–127345. Curran Associates, Inc. *equal supervision.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Najoung Kim and Tal Linzen. 2020. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105, Online. Association for Computational Linguistics.

Brenden Lake and Marco Baroni. 2018. Generalization without Systematicity: On the Compositional Skills of Sequence-to-Sequence Recurrent Networks. In *International Conference of Machine Learning (ICML 2018)*.

Karim Lasri, Tiago Pimentel, Alessandro Lenci, Thierry Poibeau, and Ryan Cotterell. 2022. Probing for the usage of grammatical number. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8818–8831, Dublin, Ireland. Association for Computational Linguistics.

Anson Lei, Bernhard Schölkopf, and Ingmar Posner. 2022. Variational causal dynamics: Discovering modular world models from interventions. *arXiv preprint arXiv:2206.11131*.

Ian Magnusson, Akshita Bhagia, Valentin Hofmann, Luca Soldaini, Ananya Harsh Jha, Oyvind Tafjord, Dustin Schwenk, Evan Pete Walsh, Yanai Elazar, Kyle Lo, Dirk Groeneveld, Iz Beltagy, Hannaneh Hajishirzi, Noah A. Smith, Kyle Richardson, and Jesse Dodge. 2024. Paloma: A benchmark for evaluating language model fit. *Preprint*, arXiv:2312.10523.

R. Thomas McCoy, Robert Frank, and Tal Linzen. 2020. Does syntax need to grow on trees? sources of hierarchical inductive bias in sequence-to-sequence networks. *Transactions of the Association for Computational Linguistics*, 8:125–140.

Moritz Miller, Bernhard Schölkopf, and Siyuan Guo. 2025. Counterfactual reasoning: an analysis of in-context emergence. *arXiv preprint arXiv:2506.05188*.

Richard Montague. 1970. Universal grammar. *Theoria*, 36(3):373–398.

Richard Montague. 1974. English as a formal language. *Formal Philosophy: Selected Papers of Richard Montague*.

Neel Nanda and Joseph Bloom. 2022. Transformerlens. `https://github.com/TransformerLensOrg/TransformerLens`.

Santiago Ontanon, Joshua Ainslie, Zachary Fisher, and Vaclav Cvicek. 2022. Making transformers solve compositional tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3591–3607, Dublin, Ireland. Association for Computational Linguistics.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Judea Pearl. 2009. *Causality*, 2 edition. Cambridge University Press.

R. Perry, J. von Kügelgen*, and B. Schölkopf*. 2022. Causal discovery in heterogeneous environments under the sparse mechanism shift hypothesis. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, pages 10904–10917. Curran Associates, Inc. *shared last author.

J. Peters, D. Janzing, and B. Schölkopf. 2017. *Elements of Causal Inference - Foundations and Learning Algorithms*. Adaptive Computation and Machine Learning Series. The MIT Press, Cambridge, MA, USA.

Andrew Radford. 1988. *Transformational Grammar: A First Course*. Cambridge Textbooks in Linguistics. Cambridge University Press.

Patrik Reizinger, Siyuan Guo, Ferenc Huszár, Bernhard Schölkopf, and Wieland Brendel. 2025a. Identifiable exchangeable mechanisms for causal structure and representation learning. In *The Thirteenth International Conference on Learning Representations (ICLR)*.

Patrik Reizinger, Bálint Mucsányi, Siyuan Guo, Benjamin Eysenbach, Bernhard Schölkopf, and Wieland Brendel. 2025b. Skill learning via policy diversity yields identifiable representations for reinforcement learning. *arXiv preprint arXiv:2507.14748*.

Jake Robertson, Arik Reuter, Siyuan Guo, Noah Hollmann, Frank Hutter, and Bernhard Schölkopf. 2025. Do-pfn: In-context learning for causal effect estimation. *arXiv preprint arXiv:2506.06039*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Sondre Wold, Lucas Georges Gabriel Charpentier, and Étienne Simon. 2025. Systematic generalization in language models scales with information entropy. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1807–1819, Vienna, Austria. Association for Computational Linguistics.

Jianyu Zhang, Niklas Nolte, Ranajoy Sadhukhan, Beidi Chen, and Léon Bottou. 2024. Memory mosaics. *arXiv preprint arXiv:2405.06394*.

A Appendix

A.1 Formalizing the dataset

Proof of Proposition 3.1. Note that $\bigcup_{k\in\{1,\ldots,R+1\}} \Omega_k \subseteq \mathcal{S}$ by definition of $(\Omega_r)_{r\in\{1,\ldots,R+1\}}$. Similarly, $\mathcal{S} \subseteq \bigcup_{k\in\{1,\ldots,R+1\}} \Omega_k$ as we *partition* the space into $R+1$ subsets. Thus, for $r \in \{1, \ldots, R\}$,

$$F_r : \Omega_r \to \left(\bigcup_{r'\in\{1,\ldots,R\}} \Omega_{r'} \right) \cup \Omega_{R+1} = \mathcal{S}.$$

□

A.2 Progression of Sentence Representations

Transformation Representation Clustering. For each transformation type (e.g., extraposition, passivization, raising), we measure the difference between mean token embeddings of transformed and original sentences from the final layer of Pythia-410M. The t-SNE visualizations (Figure 7) show how initially overlapping transformation representations become well-separated by mid-training (16K–64K steps), revealing the emergence of structured clusters for distinct syntactic operations. PCA plots (Figure 8) further capture this progression, highlighting how the model organizes syntactic variation along major linear components. Clusters of similar color correspond to transformations within the same broad category (e.g., passives, raising, movement), suggesting hierarchical organization of grammatical knowledge.

L2 Norm Progression. The L2 norms of transformation embedding differences track how syntactic representations evolve during training. Related transformations—such as passive (`np_passive_1-3`), raising (`np_raising_1-3`), and movement (`i_movement_1`, `v_movement_1-2`)—exhibit aligned trajectories, indicating coherent internal grouping. Most norms remain stable early on but rise sharply in later stages, paralleling performance gains. Across datasets (WikiText, PALOMA WikiText-103, PALOMA Penn Treebank), lower perplexity correlates with higher transformation norms, suggesting that improved language modeling coincides with more structured syntactic representations.

A.3 Comprehensive Data Summary

Our syntactic transformation dataset comprises 22,981 sentence pairs spanning 381,256 words (474,907 tokens). The dataset includes ten single-level transformations covering major syntactic operations: extraposition, movement (I-movement, V-movement), passivization (three variants), and raising constructions (three variants), totaling 19,100 examples. Additionally, we include eight nested transformation sequences that combine two operations (e.g., NP Raising + Extraposition, Passive + I Movement), contributing 3,881 examples. The nested transformations test compositional syntactic understanding by requiring models to process multiple sequential grammatical operations. Transformation counts range from 1,421 to 2,000 for single transformations and 466 to 499 for nested transformations.

A.4 Detailed Transformation Descriptions

Single Transformations Single transformations represent fundamental operations that systematically modify sentence structure. These include movement operations (Extraposition, I-Movement, V-Movement), voice alternations (NP Passive variants), and structural rearrangements (NP Raising variants). Each transformation follows specific patterns, moving constituents like noun phrases or auxiliary verbs. Examples and the patterns are shown in Table 5.

Nested Transformations Nested transformations combine two compatible single-level transformations in sequence, creating complex derivations that test multi-step grammatical operations. By requiring sequential transformations, these examples provide challenging evaluations of syntactic competence and reveal how well models track structural dependencies across derivational steps. The patterns are the same as in Table 5 and examples are provided in Table 6. Finally, our experiments focus on two-step compositions, a deliberate choice motivated by two considerations. (1) We aim to test the simplest non-trivial case of compositional syntactic generalization: if the model fails on a single step, it is unlikely to succeed on

Metric	Exact (n=2192)	Partial (n=119)
ROUGE-1	0.9920	0.8531 ± 0.1061
ROUGE-2	0.9818	0.6645 ± 0.1800
ROUGE-L	0.9903	0.8214 ± 0.1058
BERTScore-F1	0.9912	0.8385 ± 0.0896
Edit distance	0.47	8.67 ± 6.71
Exact match	94.6%	–

Table 3: Semantic and surface-level similarity between expected and generated outputs for the LoRA fine-tuned Llama3-8B. BERTScore uses `roberta-large` with baseline rescaling.

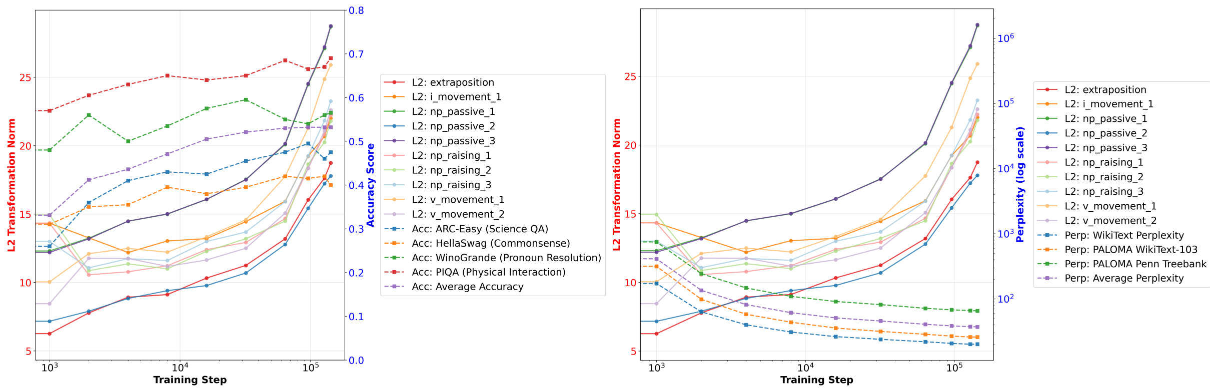


Figure 6: (Left) L2 norm progression compared to LM Harness Benchmarks. (Right) L2 norm progression compared to perplexity on WikiText-103 and Paloma Benchmarks

longer compositions. A similar pattern is observed in mathematical reasoning (Dziri et al., 2023), where compositional ability degrades with longer chains. (2) Unlike simple command-based datasets (Lake and Baroni, 2018), nesting linguistic transformations requires non-trivial, context-dependent adaptations due to idiosyncratic norms. Naively chaining transformations beyond a small number produces unlikely sentences. For example: Base: *It seems that the committee approved the proposal* (NP Raising 1) → *The committee approved the proposal* (NP Passive 1) → *The proposal was approved by the committee* (Extraposition) → *The proposal was approved by the committee of the proposal* (I Movement) → *Was the proposal approved by the committee of the proposal.*

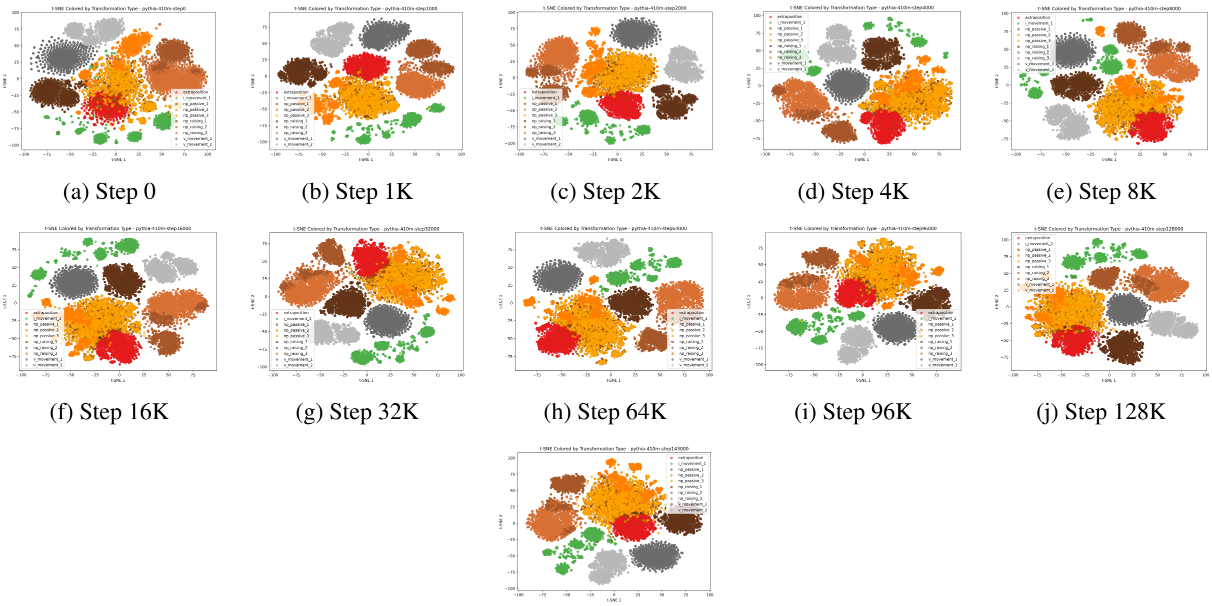
A.5 Finetuning on Transformations

OOD Transformations Between the two unseen compositions, the scores for A+C are 0.0% exact and 89.11% partial. The scores for H+E are 0.0% exact and 2.81% partial with the LoRA finetuned models with no intermediate steps.

Pythia-410M Results As scaling effects may influence the balance between memorization and generalization, we ran the fine-tuning experiments with Pythia-410M. Our results show that partial accuracy of 89.1% and exact accuracy of 81.66%. On OOD transformations, the model had 18.19% partial accuracy, which aligns with our results on Llama3-8B. Finally, we extended the dataset to at least 10,000 examples per transformation and found the exact accuracy to be 0.0% and partial accuracy to be 50.05% on OOD transformations.

Table 4: Dataset summary for single and nested transformations

Transformation	**Count**	**Total Words**	**Total Tokens**
Single-level Transformations			
Extraposition	1999	28,497	35,118
I Movement	1917	23,077	28,148
NP Passive 1	1785	24,016	29,908
NP Passive 2	1996	26,154	32,356
NP Passive 3	1421	19,222	23,967
NP Raising 1	2000	32,536	40,683
NP Raising 2	1982	31,784	39,762
NP Raising 3	2000	41,090	51,906
V Movement 1	2000	22,976	27,887
V Movement 2	2000	26,055	31,888
Total Single	**19,100**	**275,407**	**341,623**
Nested-level Transformations			
NP Raising 1 + Extraposition	466	16,752	21,135
NP Raising 3 + I Movement	472	11,952	14,946
NP Passive 3 + I Movement	467	13,851	17,622
NP Passive 2 + I Movement	495	10,500	13,327
NP Raising 1 + NP Passive 1	488	13,282	16,701
V Movement + I Movement	498	9,387	11,422
Extraposition + NP Passive 1	496	13,461	17,010
Np Raising 3 + NP Passive 3	499	16,664	21,121
Total Nested	**3,881**	**105,849**	**133,284**
Overall	**22,981**	**381,256**	**474,907**



(a) Step 0 (b) Step 1K (c) Step 2K (d) Step 4K (e) Step 8K

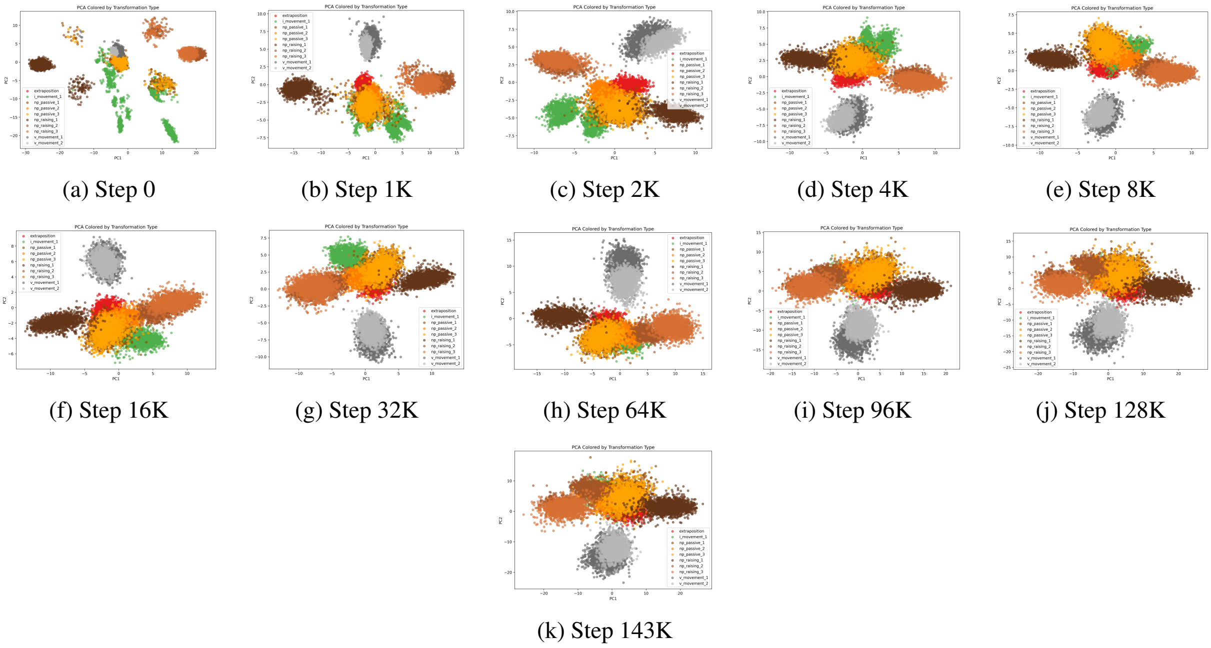
(f) Step 16K (g) Step 32K (h) Step 64K (i) Step 96K (j) Step 128K

(k) Step 143K

Figure 7: t-SNE visualization of syntactic transformation embedding differences across Pythia-410M training. Each subplot shows the 2D t-SNE projection of embedding difference vectors (transformed - original) for all transformation types at different training steps (0, 1K, 2K, 4K, 8K, 16K, 32K, 64K, 96K, 128K, 143K). Different colors represent different syntactic transformations.

Table 5: Description and examples for single-level transformations

Transformation	Description	Pattern	Example
Extraposition	Moves prepositional phrases from within noun phrases to sentence-final position	`[NP + PP][VP] → [NP][VP][PP]`	"The book on the table disappeared" → "The book disappeared on the table"
I-Movement	Moves auxiliaries/modals to sentence-initial position to form questions	`[NP][Aux/Modal][VP] → [Aux/Modal][NP][VP]?`	"She can swim" → "Can she swim?"
NP Passive 1	Standard active-to-passive with transitive verbs	`[NP subject][V][NP object] → [NP object][be + past participle][by NP subject]`	"The teacher graded the exams" → "The exams were graded by the teacher"
NP Passive 2	Small clause transformation with expanded structures	`[V][NP][small clause] → [V][that/to clause]`	"I consider him intelligent" → "I consider him to be intelligent"
NP Passive 3	Active-to-passive with clear subject-verb-object structures	`[NP subject][V][NP object] → [NP object][be + past participle][by NP subject]`	"The scientist discovered the formula" → "The formula was discovered by the scientist"
NP Raising 1	Expletive "it" structure to raised subject	`[It][verb][that[NP VP]] → [NP][verb][to VP]`	"It seems that John is happy" → "John seems to be happy"
NP Raising 2	Reverse raising: raised subject back to expletive structure	`[NP][verb][to VP] → [It][verb][that[NP VP]]`	"Mary seems to be tired" → "It seems that Mary is tired"
NP Raising 3	Complex raising with experiencer	`[NP_1][verb][to NP_2][to VP] → [It][verb][to NP_2][that[NP_1 VP]]`	"John seems to me to be honest" → "It seems to me that John is honest"
V-Movement 1	Integrates separated infinitive components	`[NP]; [VP infinitive] → [NP][VP finite]`	"The children; to play outside" → "The children play outside"
V-Movement 2	Integrates separated modal components	`[NP]; [modal]; [VP] → [NP][modal VP]`	"The students; can; solve the problem" → "The students can solve the problem"



(a) Step 0 (b) Step 1K (c) Step 2K (d) Step 4K (e) Step 8K

(f) Step 16K (g) Step 32K (h) Step 64K (i) Step 96K (j) Step 128K

(k) Step 143K

Figure 8: PCA clustering analysis of syntactic transformation embedding differences across Pythia-410M training. Each subplot shows the principal component analysis of embedding difference vectors for all transformation types at different training steps. Clustering patterns reveal how the model's internal representations of syntactic transformations evolve during training.

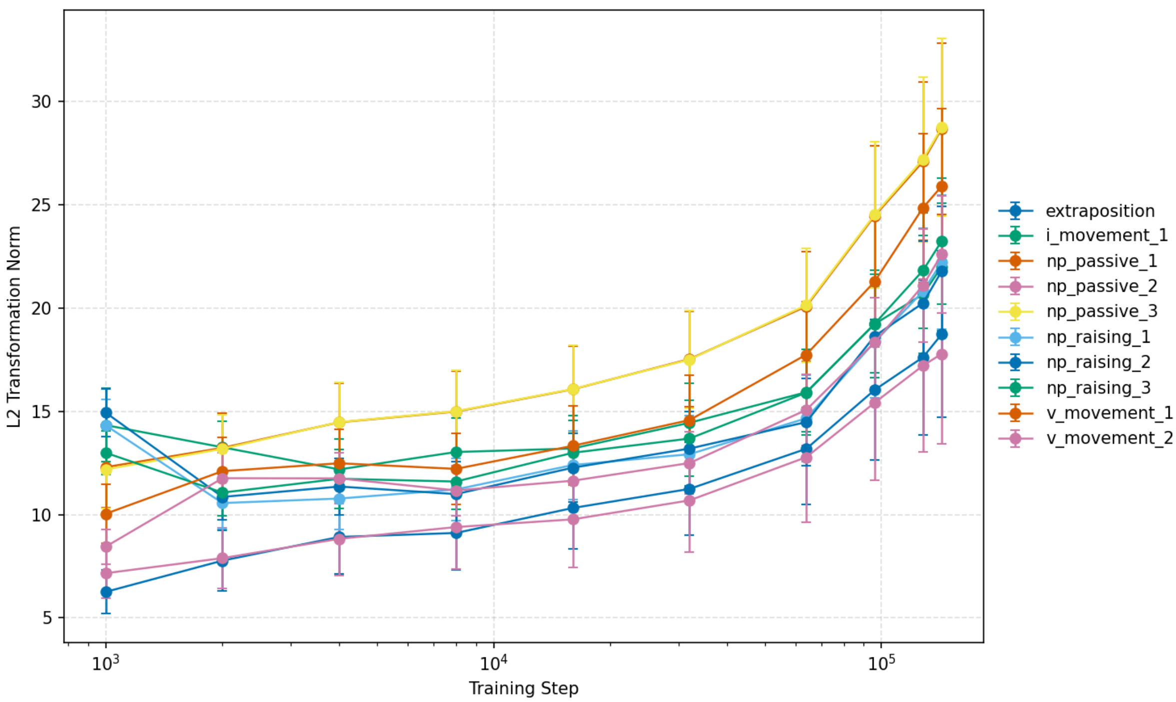


Figure 9: L2 transformation norms for Pythia-410M training checkpoints.

Table 6: Description and examples for nested transformations. Nested transformations are combinations of two compatible single-level transformations

Transformation Sequence	Example
NP Passive 2 → I Movement	"[empty] Put the corn on the table" → "The corn was put on the table" → "Was the corn put on the table?"
NP Passive 3 → I Movement	"The baker took the muffin away" → "The muffin was taken away by the baker" → "Was the muffin taken away by the baker?"
NP Raising 1 → Extraposition	"It seems that a review of my latest book appeared in the news" → "A review of my latest book appears to have appeared in the news" → "A review appears to have appeared in the news of my latest book"
V Movement → I Movement	"The children; to play outside" → "The children play outside" → "Do the children play outside?"
Extraposition → NP Passive 1	"The student from the university wrote the essay" → "The student wrote the essay from the university" → "The essay was written from the university by the student"
NP Raising 3 → NP Passive 3	"It seems that the chef placed the ingredients on the counter" → "The chef seems to place the ingredients on the counter" → "The ingredients seem to be placed on the counter by the chef"