Representational Homomorphism Error Predicts Compositional Generalization In Language Models

Zhiyu An
University of California, Merced
Merced, CA 95343
zan7@ucmerced.edu

Wan Du
University of California, Merced
Merced, CA 95343
wdu3@ucmerced.edu

Abstract

Compositional generalization—the ability to understand novel combinations of familiar components—remains a significant challenge for neural networks despite their success in many language tasks. Current evaluation methods focus on behavioral measures that reveal *when* models fail to generalize compositionally, but provide limited insight into *why* these failures occur at the representational level. We introduce *Homomorphism Error* (HE), a structural metric that quantifies how well neural network representations preserve compositional operations by measuring deviations from approximate homomorphisms between expression spaces and their internal representations. Through controlled experiments on SCAN-style synthetic compositional tasks and small-scale Transformers, we demonstrate that HE serves as a strong predictor of out-of-distribution generalization performance, achieving $R^2 = 0.73$ correlation with OOD compositional generalization accuracy. Furthermore, our analysis reveals that model size has minimal impact on compositional structure, training data coverage exhibits threshold effects, but noise injection systematically degrades compositional representations in predictable ways. These findings provide new mechanistic insights into compositional learning and establish homomorphism error as a valuable diagnostic tool for developing more robust neural architectures and training methods.

Code and data will be made publicaly available.

1 Introduction

Human language understanding is characterized by systematic compositionality—the ability to combine familiar components in novel ways to understand expressions never encountered before [5, 15]. For instance, once a person learns the meaning of "jump twice" and "turn" they can immediately comprehend "turn twice" without explicit instruction. This compositional capacity enables humans to generalize from limited experience to an infinite space of possible expressions.

Despite remarkable progress in natural language processing, modern neural networks struggle with systematic compositional generalization [3]. Empirical studies using benchmarks like SCAN [12], COGS [10], and CFQ [9] have repeatedly demonstrated that while models achieve high accuracy on training distributions, they fail catastrophically when tested on novel combinations of familiar components. This limitation poses fundamental questions about whether neural architectures can truly capture the algebraic nature of human language understanding.

Current approaches to evaluating compositional generalization primarily rely on behavioral measures—comparing model outputs against expected results on held-out test sets. While such measures reveal *when* models fail to generalize, they provide limited insight into *why* these failures occur. Understanding the internal mechanisms that support or hinder compositional reasoning requires

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

examining how models represent and manipulate compositional structure in their hidden layers, beyond surface-level performance metrics.

In this work, we introduce *Homomorphism Error* (HE), a novel structural metric that quantifies how well neural network representations preserve compositional operations. Drawing inspiration from abstract algebra, we formalize compositionality as approximate homomorphisms between expression spaces and their representations. Low homomorphism error indicates that a model's internal representations respect compositional structure—that is, the representation of a composed expression can be systematically derived from the representations of its components. High homomorphism error suggests entangled or memorization-driven representations that fail to capture underlying compositional principles.

We evaluate our approach on a customized SCAM-style synthetic dataset that allows systematic control over compositional structure, training data coverage, and noise levels, as well as building held-out test sets to measure Out-Of-Distribution (OOD) compositional generalization accuracy. Our results show that homomorphism error successfully identifies when models learn genuinely compositional representations versus when they rely on spurious correlations or memorization strategies, as shown by comparing OOD generalization accuracies and HE measurements. This structural perspective opens new methods for developing more compositionally robust neural architectures and training procedures.

Our key contributions are:

- We formalize compositionality as approximate homomorphism between syntactic and semantic algebras, and introduce homomorphism error as a task-independent metric that assesses compositional structure directly from model representations, complementing existing behavioral evaluation methods.
- Through controlled experiments on SCAM-style synthetic compositional tasks, we demonstrate that homomorphism error serves as a reliable predictor of out-of-distribution generalization performance, achieving $R^2 = 0.73$ correlation in our noise injection studies.
- Our analysis reveals that different aspects of compositionality (unary vs. binary operations) exhibit distinct sensitivities to distributional shifts in training data, providing new understanding of compositional learning mechanisms.

2 Related Work

Compositional Generalization Benchmarks. The systematic evaluation of compositional generalization in neural networks began with Lake and Baroni's introduction of the SCAN dataset [12]. SCAN demonstrated that sequence-to-sequence models, while achieving high training accuracy, failed catastrophically when tested on systematic recombinations of known components. This work established the empirical foundation for studying the systematicity challenge first articulated by Fodor and Pylyshyn [5].

Building on SCAN's foundation, subsequent benchmarks have explored different facets of compositional generalization. COGS [10] introduced semantic parsing challenges with natural language, finding that Transformers achieved near-perfect in-distribution accuracy (96-99%) but much lower out-of-distribution performance (16-35%). The grounded SCAN (gSCAN) benchmark [17] extended compositional evaluation to situated language understanding, where meaning depends on visual context. CFQ [9] provided large-scale evaluation through systematically constructed train-test splits that maximize compound divergence while minimizing atom divergence.

Theoretical Frameworks. Hupkes et al. [6] provided a comprehensive taxonomic framework, identifying five key aspects of compositionality: systematicity, productivity, substitutivity, localism, and overgeneralization. This theoretical foundation has guided much subsequent work in evaluating and understanding compositional behavior. Recent surveys [19] have further connected compositionality to broader questions of generalization and human-like reasoning in AI systems.

The field has developed several quantitative approaches to measuring compositional generalization. Keysers et al. [9] introduced compound divergence as a metric for assessing train-test split difficulty, finding strong negative correlations between compound divergence and model accuracy. Other work

has connected systematic generalization to information entropy [20], showing that generalization scales with the distributional properties of compositional components in training data.

Architectural Solutions. Various architectural innovations have been proposed to improve compositional generalization. Meta-learning approaches, particularly the MLC (Meta-Learning for Compositionality) framework [13], have shown that neural networks can achieve human-like systematicity when optimized specifically for compositional skills. Neuro-symbolic approaches like the Compositional Program Generator [11] achieve perfect performance on compositional benchmarks with dramatically improved sample efficiency.

For Transformer architectures, improvements have come through auxiliary training objectives [8], curriculum learning with dataset cartography [7], and architectural modifications such as increased depth [14]. Graph-based semantic parsing frameworks have shown particular promise for structural generalization tasks [16].

Internal Representation Analysis. Understanding the internal mechanisms underlying compositional behavior has been addressed through various probing methodologies [1]. Work in mathematical reasoning has demonstrated that neural networks can learn compositionally structured representations that reflect sub-expression meanings [18]. Recent neuroscience-inspired work has shown evidence for compositional representations through algebraic operations on brain activity patterns [4].

However, existing approaches to measuring compositionality have primarily focused on behavioral evaluation or task-specific probing. Our homomorphism error metric differs by providing a principled, architecture-agnostic measure of how well models preserve compositional structure in their internal representations, independent of surface-level task performance. This structural approach offers new insights into the mechanistic basis of compositional generalization failures and successes.

3 Homomorphism Error as Structural Metric for Compositionality

We begin by formalizing compositionality in terms of homomorphisms between syntactic expressions and their semantic interpretations. Let $\mathcal{P}$ denote a finite set of primitives and $\circ$ a syntactic composition operator defined by a grammar G. Let $\mathcal{E}$ be the set of expressions generated from $\mathcal{P}$ using $\circ$. Each expression $e \in \mathcal{E}$ has an associated semantic interpretation $[\![e]\!] \in \mathcal{S}$, where $(\mathcal{S}, \bullet)$ is a semantic algebra with composition operator $\bullet$.

Compositionality as homomorphism. We say the mapping $[\![\cdot]\!] : \mathcal{E} \to \mathcal{S}$ is *compositional* if for all $e_1, e_2 \in \mathcal{E}$,

$$[\![e_1 \circ e_2]\!] \;=\; [\![e_1]\!] \bullet [\![e_2]\!]. \tag{1}$$

That is, the meaning of a composed expression is given by the composition of the meanings of its parts.

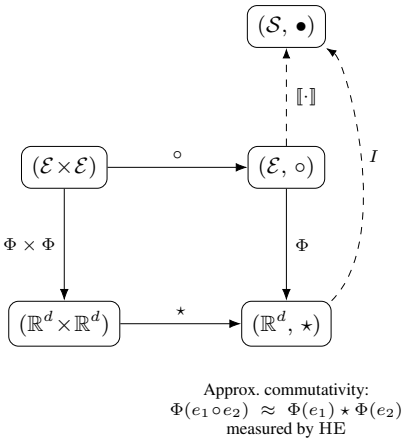

Figure 1: Compositionality as approximate homomorphism. Solid arrows form the representation-level square; dashed arrows show semantic interpretation $[\![\cdot]\!]$ and an evaluation/readout I. Homomorphism Error (HE) quantifies how well Φ preserves composition.

Approximate homomorphism in language models. Consider a language model M_θ with hidden representation function $\Phi_\ell : \mathcal{E} \to \mathbb{R}^d$ at layer ℓ. We introduce an auxiliary learnable operator $\star : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ (e.g. linear map, bilinear map, or MLP) trained to approximate compositionality at the representation level. The *Homomorphism Error (HE)* at layer ℓ is

$$\mathrm{HE}_\ell \;=\; \mathbb{E}_{(e_1,e_2)\sim\mathcal{D}}\Big[d\big(\Phi_\ell(e_1 \circ e_2),\; \Phi_\ell(e_1) \star_\ell \Phi_\ell(e_2)\big)\Big], \tag{2}$$

where $\mathcal{D}$ is a distribution over expressions and d is a distance metric such as mean squared error (MSE). In practice, we extract pairs of compossible expressions from the training dataset where $\star_\ell$ learns to predict the representation of a composed expression from its components. We instantiate $\star_\ell$

with three operator families (linear, bilinear, MLP) and report the average error across them to avoid biasing the analysis toward a particular functional form of composition.

Interpretation. Low HE indicates that internal representations are *approximately homomorphic* with respect to the task's compositional structure, whereas high HE suggests entangled or memorization-driven representations. Thus, HE measures the extent to which compositional structure is linearly or non-linearly decodable from hidden states, independent of task accuracy.

4 Experiment

4.1 Dataset Construction

We design a controlled synthetic dataset inspired by SCAN-style tasks in order to probe compositional generalization. Let $\mathcal{P}$ denote a finite set of *primitives* (e.g., `walk`, `jump`, `look`, `turn`) that each map to atomic output sequences over a target alphabet Σ. We further introduce a set $\mathcal{M}$ of *modifiers* (e.g., `twice`, `thrice`) that act as unary operators on primitives, and a set $\mathcal{C}$ of *connectors* (e.g., `then`) that define binary composition.

Formally, the grammar G for input expressions is defined as

$$e \;::=\; p \;\mid\; m(e) \;\mid\; e_1\, c\, e_2, \quad \text{where } \; p \in \mathcal{P}, m \in \mathcal{M}, c \in \mathcal{C}.$$

The semantics $[\![e]\!]$ is an output sequence in Σ^* defined compositionally by rules such as

$$[\![m(e)]\!] \;=\; f_m([\![e]\!]), \qquad [\![e_1\, c\, e_2]\!] \;=\; g_c([\![e_1]\!], [\![e_2]\!]),$$

where f_m and g_c are deterministic rewriting functions. For example,

$$[\![\texttt{jump twice}]\!] = [\![\texttt{jump}]\!]\,[\![\texttt{jump}]\!], \qquad [\![\texttt{look then walk}]\!] = [\![\texttt{look}]\!]\,[\![\texttt{walk}]\!].$$

To study the effect of different amount of noise in the training dataset, we introduce a finite set of *noise tokens* denoted by $\mathcal{K}$ (e.g. `foo`, `bar`, `baz`). When constructing noisy datasets, the noise tokens are inserted at random positions in the prompts, but not in the outputs. For example,

$$\underbrace{\texttt{foo jump bar thrice then look baz}}_{\text{input}} \mapsto \underbrace{\texttt{jump jump jump look}}_{\text{output}}$$

This dataset construction allows us to systematically control the number of primitives, modifiers, connectors, and noise tokens during training, and to evaluate generalization to held-out combinations.

4.2 Homomorphism Error Probe Design

Building on the general definition of Homomorphism Error (HE) in Section 3, we distinguish two forms of compositionality present specifically in the above dataset, namely *modifier HE* and *sequence HE*, by the number of parameters (unary/binary) that the compositional operation requires.

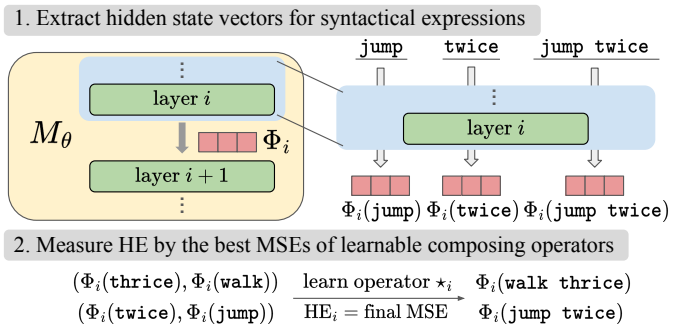

Figure 2: Illustration of the HE measuring procedure.

Modifier HE. Modifier homomorphism concerns unary composition $m(e)$, such as `twice` and `thrice`. For a representation function Φ at model layer ℓ, we define

$$\mathrm{HE}_\ell^{\mathrm{mod}} \;=\; \mathbb{E}_{(m,e)\sim\mathcal{D}_{\mathrm{mod}}}\Big[d\big(\Phi_\ell(m(e)),\; \star_\ell^m(\Phi_\ell(e))\big)\Big], \tag{3}$$

where $\star_\ell^m : \mathbb{R}^d \to \mathbb{R}^d$ is a learned operator specific to modifier m, and d is a distance metric such as MSE. Low $\mathrm{HE}^{\mathrm{mod}}$ indicates that modifiers are represented as structure-preserving transformations.

Sequence HE. Sequence homomorphism concerns binary composition $e_1 \, c \, e_2$. For representation function Φ_ℓ, we define

$$\mathrm{HE}_\ell^{\mathrm{seq}} \;=\; \mathbb{E}_{(e_1,c,e_2)\sim\mathcal{D}_{\mathrm{seq}}}\Big[d\big(\Phi_\ell(e_1 \, c \, e_2),\; \star_\ell^c(\Phi_\ell(e_1), \Phi_\ell(e_2))\big)\Big], \tag{4}$$

where $\star_\ell^c : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ is a learned binary operator associated with connector c. Low $\mathrm{HE}^{\mathrm{seq}}$ indicates that connectors are represented as structure-preserving composition operators.

When calculating Modifier HE, we extract (primitive, modifier, combined) triples from the training dataset, where the combined representation is the mean pooling of consecutive primitive and modifier token representations. For Sequence HE, we extract (part1, part2, combined) triples where parts are primitive-initiated segments and the combined representation is their average. Noise tokens are not compossible with any other tokens and are not included for HE calculation, thus only the HEs of meaningful compositions are measured.

4.3 Experiment Designs

We conduct three families of controlled experiments to investigate how model architecture, training data composition, and noise affect both out-of-distribution (OOD) generalization and internal compositional structure as measured by the homomorphism errors. All experiments used a fixed OOD test sets composed of held-out expressions containing 5 to 12 primitives. 200 unique expressions were sampled from the space of primitives with each number of primitives. Layer-wise $\mathrm{HE}^{\mathrm{mod}}$ and $\mathrm{HE}^{\mathrm{seq}}$ are computed to determine whether deeper models learn more compositional internal representations.

Model architecture and training. All models are decoder-only transformers trained with a causal language modeling objective. All experiments used a set of fixed hyperparameters: hidden dimension $d = 128$, number of attention heads $h = 4$, feedforward dimension 256. Inputs are tokenized and passed through learned embeddings with positional encodings. The final hidden states are projected to the vocabulary space via a linear output layer. Models are trained using cross-entropy loss with teacher forcing, optimized with Adam ($\beta_1 = 0.9, \beta_2 = 0.98$), learning rate 10^{-4}, and batch size 64. Training runs for 50 epochs with early stopping on validation loss. All experiments share the same optimization settings to isolate the effects of model depth, training sparsity, and noise.

1. Model size ablation. We vary the number of transformer layers $L \in \{1, 2, \ldots, 10\}$. For each configuration, we train models on a fixed dataset containing 2 primitives with no noise.

2. Training data sparsity. To probe the effect of training data coverage, we construct datasets with the expressions containing increasing numbers of primitives: $1, 2, 3, 4$, always keeping $\mathrm{num\_noise} = 0$. For each sparsity level, models are trained with a fixed architecture ($L = 4$ layers) and evaluated on the same OOD test sets as above. This experiment tests whether reduced training coverage leads to higher homomorphism error and worse generalization.

3. Noise injection. We evaluate the robustness of learned compositional representations to spurious tokens by constructing training datasets with 2 primitives and varying numbers of randomly inserted noise tokens $\mathrm{num\_noise} \in \{0, 1, \ldots, 15\}$. Models are trained with 4 layers and evaluated on OOD sequences without noise. Both accuracy and homomorphism error are tracked to determine whether noise disrupts compositional representations.

Evaluation metrics. For each experiment, we report:

- **OOD accuracy:** average fraction of correctly predicted output sequences on held-out test sets.
- **Modifier HE:** layer-wise MSE between representations of $m(e)$ and $\star_\ell^m(\Phi_\ell(e))$ across all modifiers in the dataset. Values are final MSEs averaged across linear, bilinear, and MLP operators.
- **Sequence HE:** layer-wise MSE between representations of $e_1 \, c \, e_2$ and $\star_\ell^c(\Phi_\ell(e_1), \Phi_\ell(e_2))$ across all connectors. Values are final MSEs averaged across linear, bilinear, and MLP operators.

Seed averaging and error bars. Each experiment is repeated with 5 random seeds to control for stochasticity in dataset generation, model initialization, and training. Reported metrics include mean and standard deviation across seeds, which are visualized as error bars in all plots.

Computing Resources. All experiments (including 5 random seeds per configuration across all ablations) were run on a single Apple M1 chip with 8GB of memory. The full experiment suite completed in approximately 30 minutes.

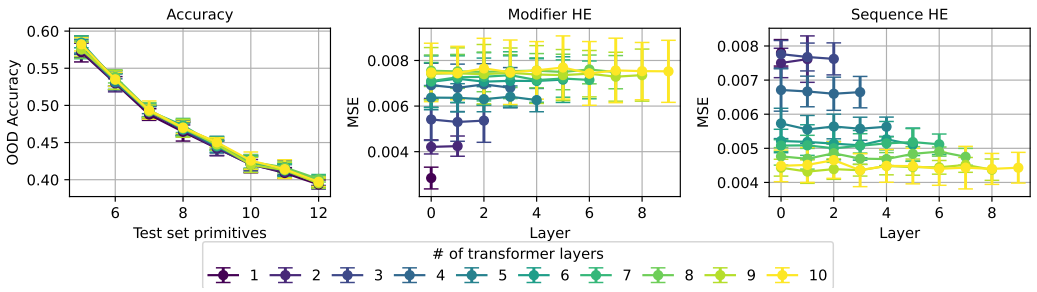

Figure 3: Model size ablation results. Lines represent number of transformer layers $L \in \{1, 2, \ldots, 10\}$ in the language model.

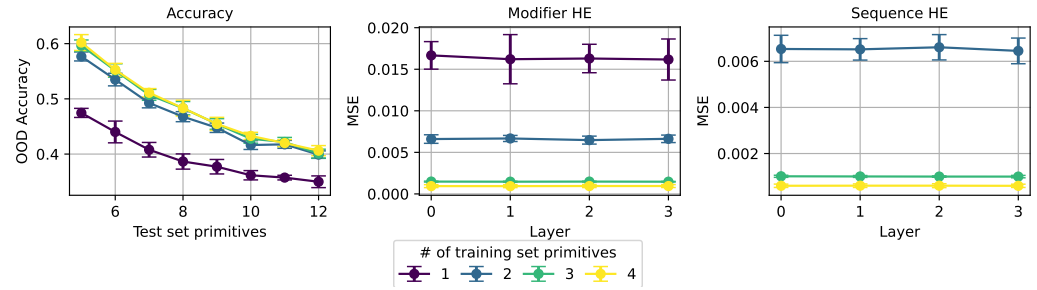

Figure 4: Training data sparsity results. Lines represent number of primitives in the expressions that the trainig set contains up-to. 4 meaning the training set contains expressions with up-to 4 primitives.

4.4 Experiment Results

Our experiments reveal three key findings about the relationship between model architecture, training data composition, compositional structure, and out-of-distribution generalization.

Model size has minimal impact on compositional generalization. Figure 3 demonstrates that increasing model depth from 1 to 10 transformer layers yields negligible improvements in OOD accuracy, with all configurations achieving approximately 40–60% accuracy across different test complexities. More importantly, both modifier and sequence homomorphism errors remain remarkably stable across model sizes, with variations on the order of 10^{-3}. The modifier HE shows slight fluctuations between layers but no systematic trend, while sequence HE exhibits similarly minimal variation. This suggests that for our controlled compositional task, representational capacity beyond a single layer provides little benefit for learning compositional structure, and that the fundamental challenge lies not in model expressivity but in the compositional inductive biases encoded during training.

Training data coverage exhibits threshold effects on compositional learning. The training data sparsity results in Figure 4 reveal a sharp threshold effect in compositional generalization. Models trained with only 1 primitive achieve significantly lower OOD accuracy (35–47%) and substantially higher modifier HE (0.015–0.018 MSE) compared to models trained with 2 or more primitives (40–60% accuracy, 0.005–0.007 MSE modifier HE). This dramatic performance gap occurs because training sets with a single primitive cannot cover binary connectors, leaving models unable to learn compositional rules for sequence construction. However, once the training set includes sufficient coverage to span the full compositional domain (2+ primitives), the marginal benefit of additional primitives rapidly diminishes. Models trained with 2, 3, or 4 primitives show nearly identical OOD accuracy and homomorphism error profiles across layers, indicating that compositional generalization depends primarily on *structural coverage* rather than *data volume*.

Noise injection systematically degrades compositional representations. The most striking results emerge from the noise injection experiment (Figure 5). Unlike model size and training data sparsity,

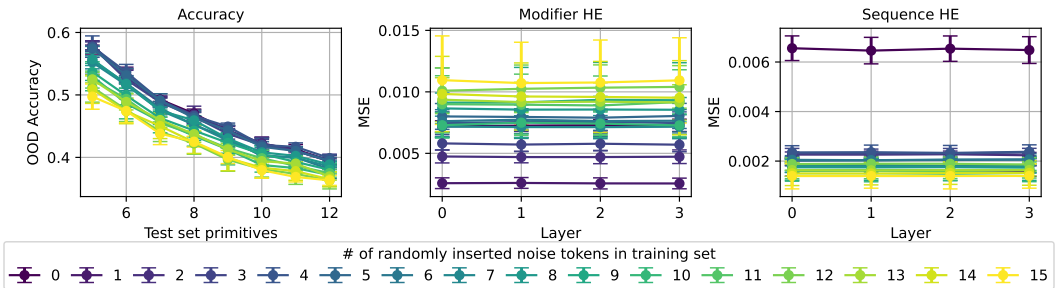

Figure 5: Noise injection results. Lines represent number of noise tokens inserted in training data.

the number of randomly inserted noise tokens exhibits a strong, monotonic relationship with both generalization performance and internal compositional structure. Models trained with increasing noise levels (0–15 tokens) show consistently degraded mean OOD accuracy, declining from approximately 47% with no noise to 42% with 15 noise tokens. This degradation is accompanied by a systematic increase in modifier HE from approximately 0.002 MSE to 0.012 MSE, while sequence HE remains relatively stable across noise levels.

The predictive power of our homomorphism error metric is most clearly demonstrated in Figure 6, which plots the relationship between mean modifier HE and mean OOD accuracy across all noise conditions. Polynomial regression analysis reveals a highly reliable relationship, with $R^2 = 0.73$ for both quadratic and cubic fits. This strong correlation indicates that modifier HE serves as an effective predictor of out-of-distribution compositional generalization, capturing the degree to which models represent modifiers as structure-preserving transformations rather than memorized input-output mappings.

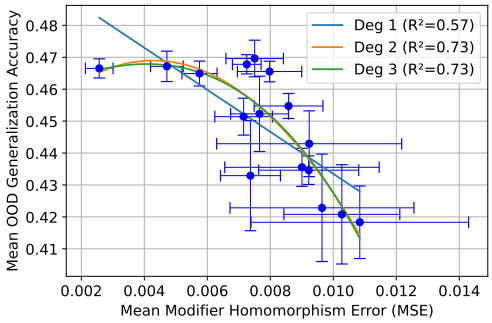

Figure 6: Analysis result of the noise injection experiment. Correlation between mean OOD generalization accuracy and mean modifier HE is shown. Polynomial regression with vaious degrees is conducted and R^2 is reported.

These results collectively suggest that compositional generalization in language models depends less on raw model capacity or training set size, more on the *structural integrity* of learned representations. Noise injection appears to interfere specifically with the model's ability to learn modifier operations as compositional functions, while preserving sequence-level compositional structure. This dissociation provides evidence that different aspects of compositionality (unary vs. binary operations) may be learned through distinct mechanisms and exhibit different sensitivities to distributional shifts in training data.

5 Discussion and Future Work

Our results provide several key insights into the mechanisms underlying compositional generalization in neural networks. The strong predictive relationship between homomorphism error and out-of-distribution performance ($R^2 = 0.73$) suggests that compositional failures are fundamentally rooted in representational structure rather than surface-level pattern matching. This finding supports theories that emphasize the importance of algebraic structure in neural representations [5] and provides empirical evidence for the homomorphism perspective on compositionality.

The dissociation between modifier and sequence homomorphism errors reveals that different aspects of compositional structure are learned through distinct mechanisms. Our noise injection experiments show that spurious tokens primarily disrupt the learning of modifier operations (unary functions) while leaving sequence composition (binary operations) relatively intact. This suggests that unary and

binary compositional operations may rely on different neural circuits or learning dynamics, opening new avenues for targeted architectural interventions.

Interestingly, our finding that model depth has minimal impact on compositional generalization challenges common assumptions about the relationship between representational capacity and systematic generalization. Instead, our results point to the quality of compositional structure in representations as the critical factor, rather than raw model expressivity.

5.1 Limitations

Several limitations constrain the generalizability of our current findings. First, our experiments are conducted on synthetic data with precisely controlled compositional structure. While this enables rigorous analysis of the homomorphism error metric, real-world language presents additional complexities including semantic ambiguity, context dependence, and irregular constructions that may not conform to strict compositional principles.

Second, our evaluation focuses on relatively small Transformer models with controlled architectures and hyperparameters. The behavior of homomorphism error in large-scale pretrained language models remains an open question, particularly given evidence that scale can partially overcome compositional limitations [2].

Third, our current framework assumes discrete, well-defined compositional operations. Extending the homomorphism error framework to capture more nuanced forms of compositionality—such as semantic composition in natural language where meaning is not strictly algebraic—presents both theoretical and computational challenges.

5.2 Future Directions

A natural next step is applying homomorphism error analysis to established compositional benchmarks using natural language, including SCAN, COGS, and CFQ. This would validate the metric's utility beyond synthetic settings and potentially reveal why certain architectural innovations succeed where others fail. Investigating how homomorphism error scales with model size and pretraining data could provide insights into whether the compositional capabilities of large language models emerge from improved representational structure or alternative mechanisms.

Furthermore, our predictive framework opens possibilities for compositionally-aware architecture search and training procedures. Future work could explore using homomorphism error as an optimization objective or regularization term, directly encouraging models to learn structured representations during training rather than discovering compositional failures post-hoc.

6 Conclusion

We introduced homomorphism error as a structural metric that formalizes compositionality as approximate homomorphisms between expression spaces and their representations. Our results demonstrate that homomorphism error reliably predicts out-of-distribution performance ($R^2 = 0.73$), revealing that representational structure, not surface-level pattern matching, underlies compositional failures. The dissociation between modifier and sequence homomorphism errors shows that different compositional aspects are learned through distinct mechanisms, with unary operations particularly vulnerable to spurious correlations. Crucially, our findings show that the quality of learned structure—rather than raw model capacity or architecture—determines systematic generalization capability. This framework provides both theoretical insights into the systematicity debate and practical diagnostic tools for developing more compositionally robust neural architectures.

References

[1] Yonatan Belinkov. Probing classifiers: Promises. *Shortcomings, and Advances*, 2021.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[3] Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, et al. Faith and fate: Limits of transformers on compositionality. *Advances in Neural Information Processing Systems*, 36:70293–70332, 2023.

[4] Matteo Ferrante, Tommaso Boccato, Nicola Toschi, and Rufin VanRullen. Evidence for compositionality in fmri visual representations via brain algebra. *Communications Biology*, 8(1):1263, 2025.

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

[6] Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795, 2020.

[7] Osman Batur Ince, Tanin Zeraati, Semih Yagcioglu, Yadollah Yaghoobzadeh, Erkut Erdem, and Aykut Erdem. Harnessing dataset cartography for improved compositional generalization in transformers. *arXiv preprint arXiv:2310.12118*, 2023.

[8] Yichen Jiang and Mohit Bansal. Inducing transformer's compositional generalization ability via auxiliary sequence prediction tasks. *arXiv preprint arXiv:2109.15256*, 2021.

[9] Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, et al. Measuring compositional generalization: A comprehensive method on realistic data. *arXiv preprint arXiv:1912.09713*, 2019.

[10] Najoung Kim and Tal Linzen. Cogs: A compositional generalization challenge based on semantic interpretation. *arXiv preprint arXiv:2010.05465*, 2020.

[11] Tim Klinger, Qi Liu, Maxwell Crouse, Soham Dan, Parikshit Ram, and Alexander G Gray. Compositional program generation for systematic generalization. In *International Joint Conference on Artificial Intelligence 2023 Workshop on Knowledge-Based Compositional Generalization*, 2023.

[12] Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International conference on machine learning*, pages 2873–2882. PMLR, 2018.

[13] Brenden M Lake and Marco Baroni. Human-like systematic generalization through a meta-learning neural network. *Nature*, 623(7985):115–121, 2023.

[14] William Merrill and Ashish Sabharwal. A little depth goes a long way: The expressive power of log-depth transformers. *arXiv preprint arXiv:2503.03961*, 2025.

[15] Richard Montague et al. Universal grammar. *1974*, pages 222–46, 1970.

[16] Alban Petit, Caio Corro, and François Yvon. Structural generalization in cogs: Supertagging is (almost) all you need. *arXiv preprint arXiv:2310.14124*, 2023.

[17] Laura Ruis, Jacob Andreas, Marco Baroni, Diane Bouchacourt, and Brenden M Lake. A benchmark for systematic generalization in grounded language understanding. *Advances in neural information processing systems*, 33:19861–19872, 2020.

[18] Jacob Russin, Roland Fernandez, Hamid Palangi, Eric Rosen, Nebojsa Jojic, Paul Smolensky, and Jianfeng Gao. Compositional processing emerges in neural networks solving math problems. In *CogSci... Annual Conference of the Cognitive Science Society. Cognitive Science Society (US). Conference*, volume 2021, page 1767, 2021.

[19] Sania Sinha, Tanawan Premsri, and Parisa Kordjamshidi. A survey on compositional learning of ai models: Theoretical and experimental practices. *arXiv preprint arXiv:2406.08787*, 2024.

[20] Sondre Wold, Lucas Georges Gabriel Charpentier, and Étienne Simon. Systematic generalization in language models scales with information entropy. *arXiv preprint arXiv:2505.13089*, 2025.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Yes.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Yes.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [NA]

   Justification: No theoretical contribution that requires proof.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Yes, will be fully opensourced.

   Guidelines:

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

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

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

   Answer: [Yes]

   Justification: Yes, will be fully opensourced.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Yes, will be fully opensourced.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Yes.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Yes, compute resources are reported.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Yes.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [NA]

    Justification: No societal impact is expected.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: No data or model with high risk of misuse.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: Yes, creators and original owners of assets are properly credited, license and terms of use are properly respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: Yes, well-documented and provided alongside the assets.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: No crowdsourcing.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: No human subject is involved.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [NA]

Justification: LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research.

Guidelines:

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