CS 957, System-2 AI
Compositionality

Mahdieh Soleymani

May 2025

Sharif University of Technology

What is compositionality?

- The principle that complex expressions derive meaning from their parts and the rules used to combine them.

- Example:
  "red ball" = meaning of "red" + meaning of "ball" + how adjectives modify nouns.

Compositionality & Human Cognition

- A fundamental characteristic common to both human vision and natural language is their compositional nature.
- An active problem in the AI community that is crucial for advancing Artificial General Intelligence (AGI)
- Compositional generalization is critical to simulate the compositional capability of humans

Why It Matters for Generalization

- Compositional structure allows:
  - **Systematic generalization**: Understanding new combinations of known elements (e.g., "green dog", "flying car").
  - **Data efficiency**: Learn more from less data by reusing parts.
  - **Robust transfer**: Apply knowledge to unseen contexts or domains.

Generalization beyond the Training Distribution

- **In-Distribution (ID) Generalization**: Test distribution is the same as training distribution
  - Learning models usually only deal with generalization within the same distribution

- SOTA AI systems learn but do not generalize well to modified distributions

Out-of-Distribution (OOD) Generalization

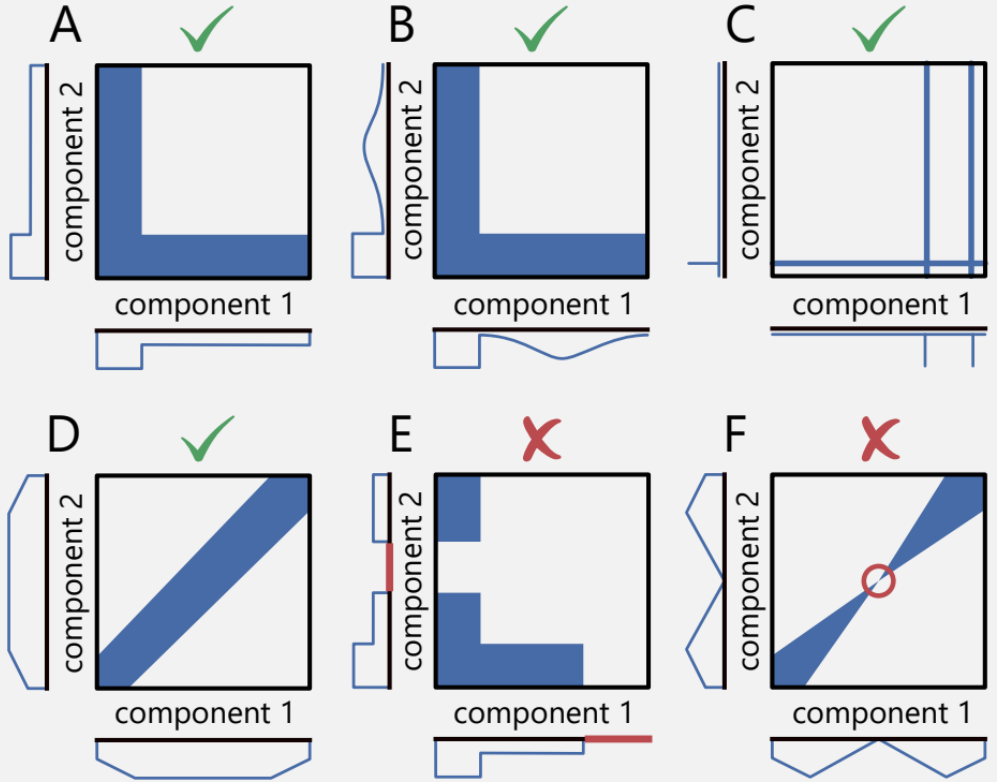
- **OOD generalization** problem for non-linear models where **train and test distributions** differ in their densities, but not their supports

- OoD generalization is a way toward more autonomous agents
  - Different places, times, sensors, actuators, goals, policies, etc.
  - Agent faces non-stationaritires
  - Changes in distribution due to their or other agents actions.

Compositional Generalization

- Compositional generalization requires generalizing to a distribution with different, possibly non-overlapping support.
  - This problem is more challenging and remains unsolved.

- Compositional Support



Wiedemer et al., Compositional Generalization from First Principles, NeurIPS 2023

Conscious Processing and OOD Setting

- Faced with novel or rare situations, humans combine on-the-fly appropriate piece of knowledge, to reason with them and imagine solutions.
- We do not follow out habitual routines, we think hard to solve new problems.
- Systematic Generalization: Inductive biases should be included in order to go to strong OOD generalization
  - Decomposing knowledge and recomposing dynamically as needed

A. Goyal and Y. Bengio, “Inductive Biases for Deep Learning of Higher-Level Cognition,” 2020

OOD Generalization in Seq-to-Seq Models

- Length Generalization
  - Ability to generalize to longer sequences than those seen during training.

- Compositional Generalization
  - Ability to generalize to combinations not seen during training.

Compositional Generalization of Seq-to-Seq Models

- Several benchmarks such as SCAN (Lake and Baroni, 2018) have been proposed based on semantic parsing tasks.
  - The training set cover all the primitives while lacking certain combinations
  - The test set focuses on these missing combinations.

- By fine-tuning generic models on these benchmarks, much work reported that these models exhibit **poor compositional generalization**

Scan Task

- SCAN task (Lake & Baroni 2018)
  - Goal: Test how well models generalize compositionality in seq-to-seq tasks
  - Task: Simple instructions mapped to action sequences
    - "turn left”-> LTURN
    - "walk forward”-> WALK
    - “walk twice”-> WALK WALK

- Objective: Train models to generalize compositionality, combining familiar components (e.g., actions, directions) to create new sequences.

- Jump Split: Generalization to new combinations
  - **Test**: New commands like jump twice and jump around

Compositional Generalization on Scan Task

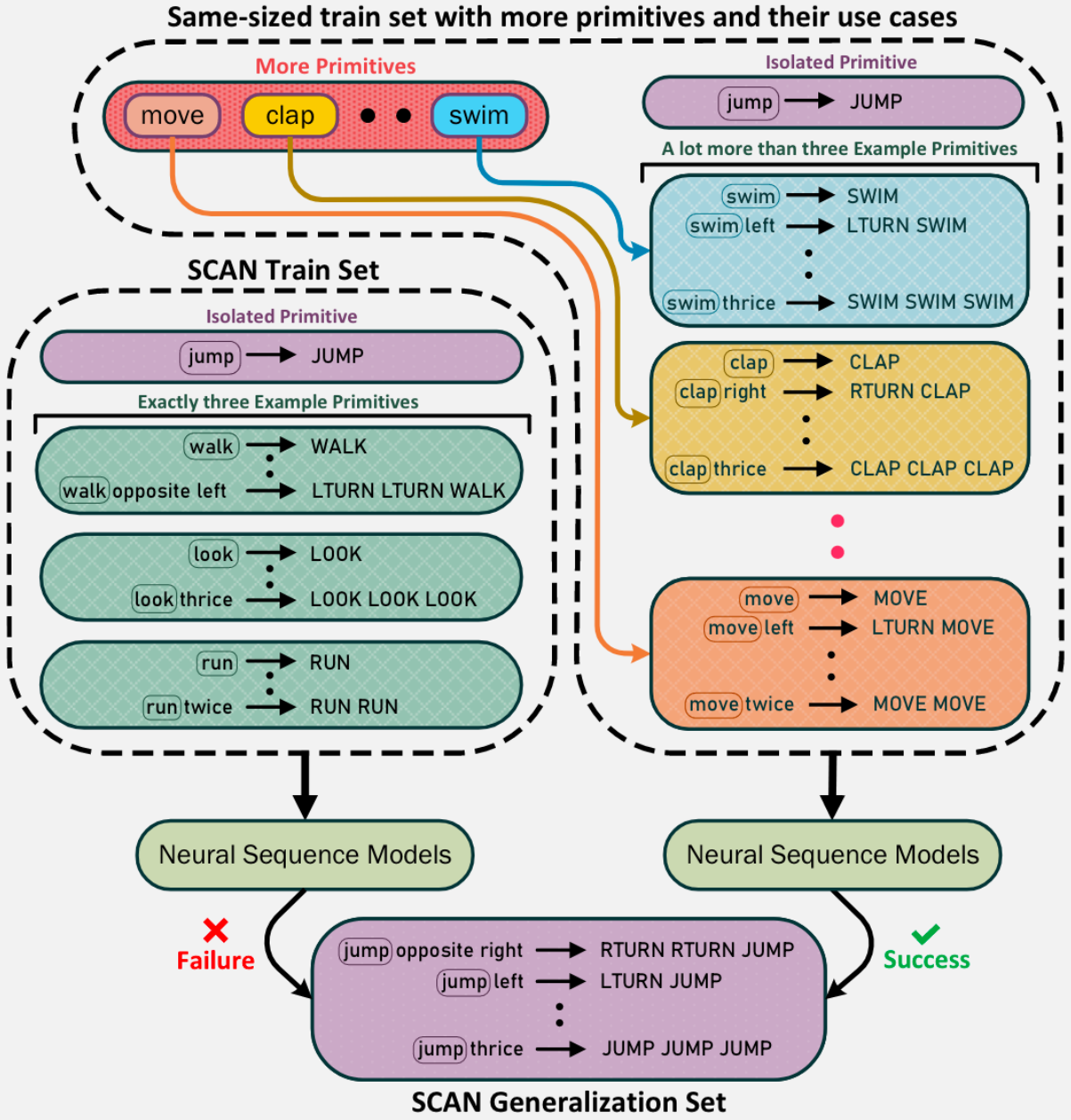
- Many studies have suggested that seq-to-seq models lack generalization on this task since they achieve near-zero accuracies

- By making simple and intuitive changes to the training data distribution, seq-to-seq models can achieve high generalization
  - if we incorporated examples with more novel primitives in the training set without necessarily increasing the size of the training set
  - even with a training set of size less than 20% of the original training set

Patel et al., Revisiting the Compositional Generalization Abilities of Neural Sequence Models, ACL 2022

Adding More Primitives

- Adding new primitives to the language and follow the same grammar rules as other existing primitives



Patel et al., Revisiting the Compositional Generalization Abilities of Neural Sequence Models, ACL 2022

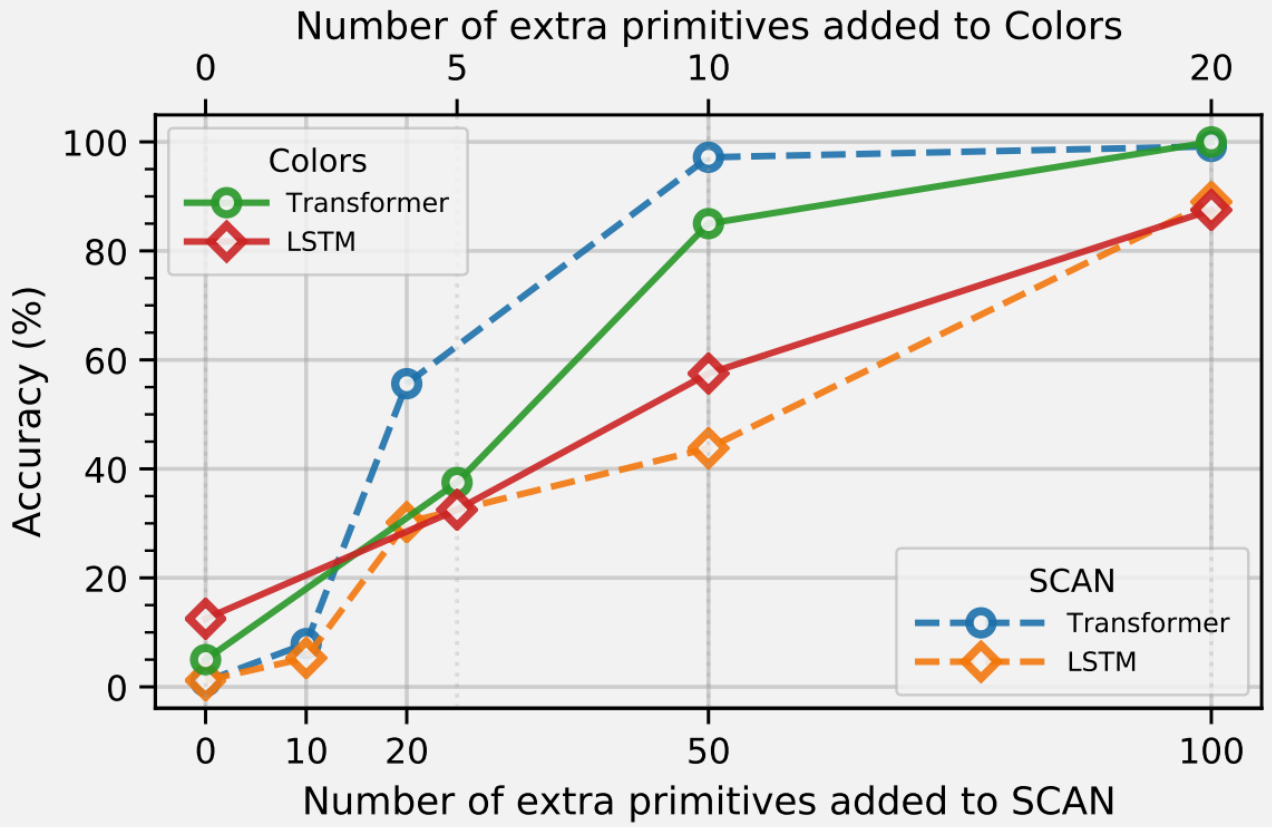


Figure 2: Generalization performance (↑) on SCAN and Colors improves with higher number of example primitives in the training set.

Patel et al., Revisiting the Compositional Generalization Abilities of Neural Sequence Models, ACL 2022

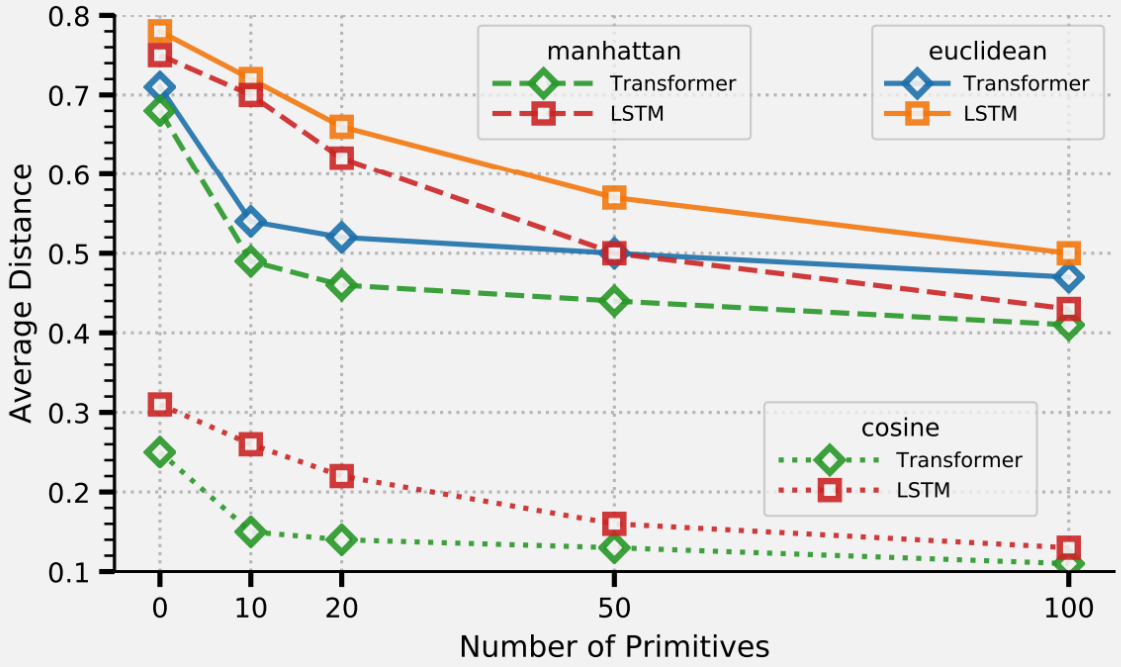


Figure 3: Measuring the distance of embedding of *isolated primitive* with embeddings of example primitives for learned Transformer and LSTM models as we increase the number of example primitives in SCAN.

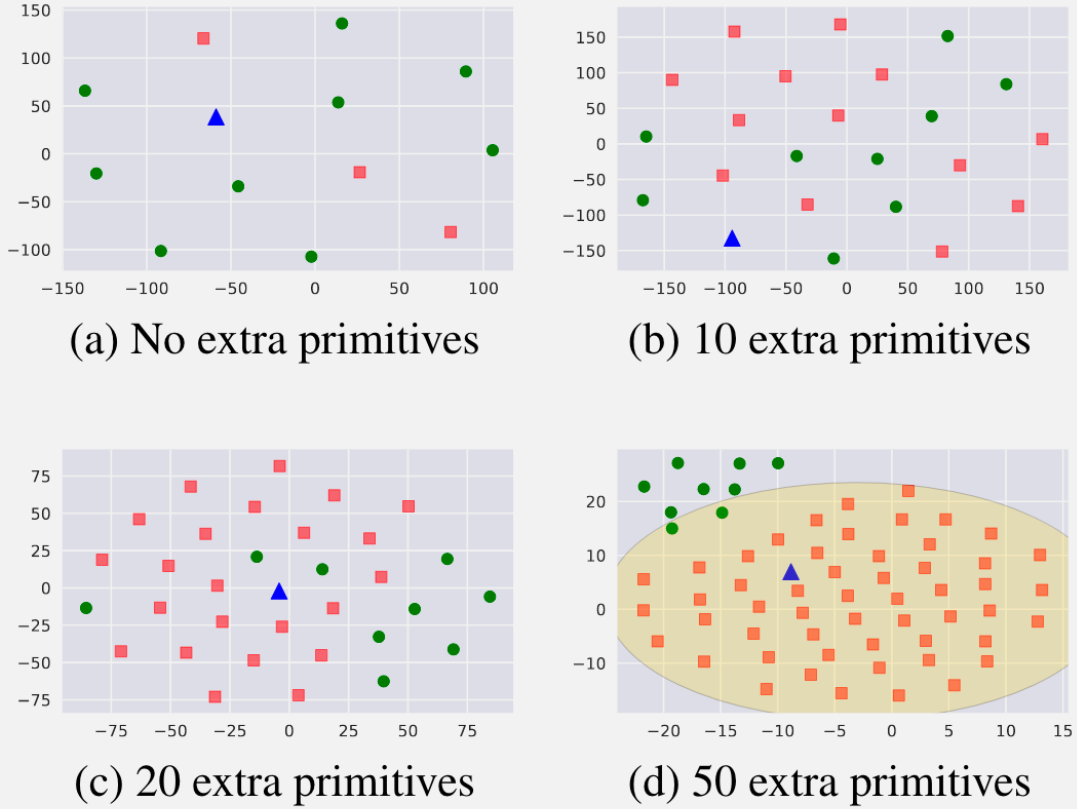
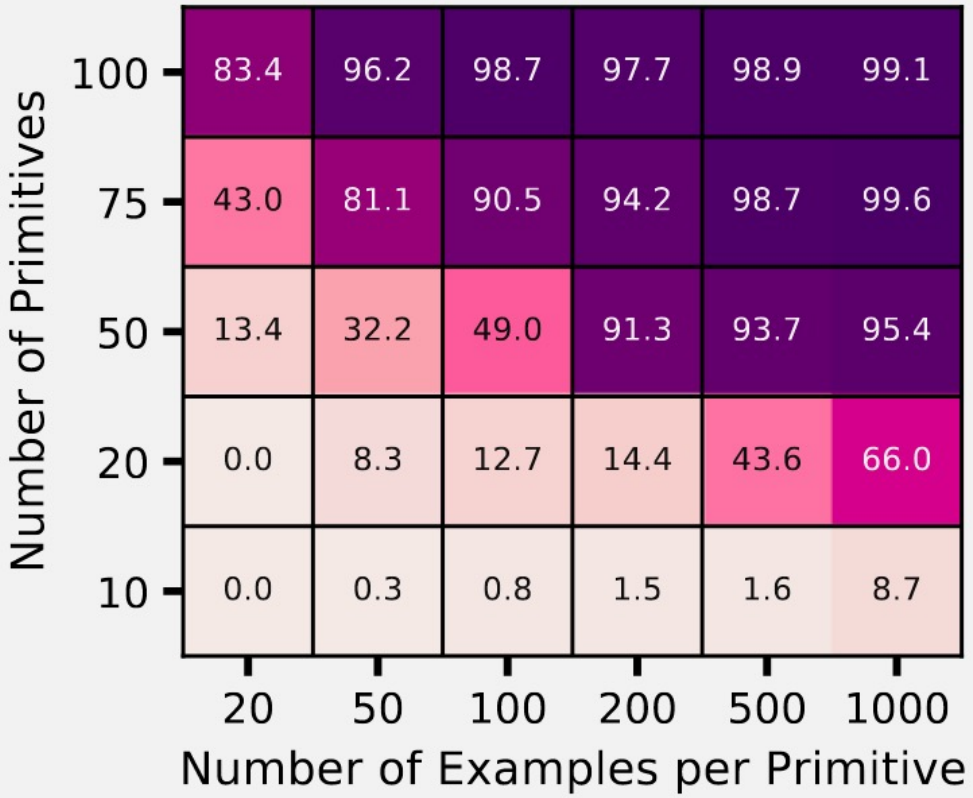


Figure 4: Visualizing the t-SNE reduced embeddings of isolated primitive (▲), example primitives (■) and non-primitives (●) from a learned Transformer model as we increase number of example primitives in SCAN.

Patel et al., Revisiting the Compositional Generalization Abilities of Neural Sequence Models, ACL 2022

Changing Training Set Distributions of the SCAN Dataset



Patel et al., Revisiting the Compositional Generalization Abilities of Neural Sequence Models, ACL 2022

Compositional Generalization vs. Model Size

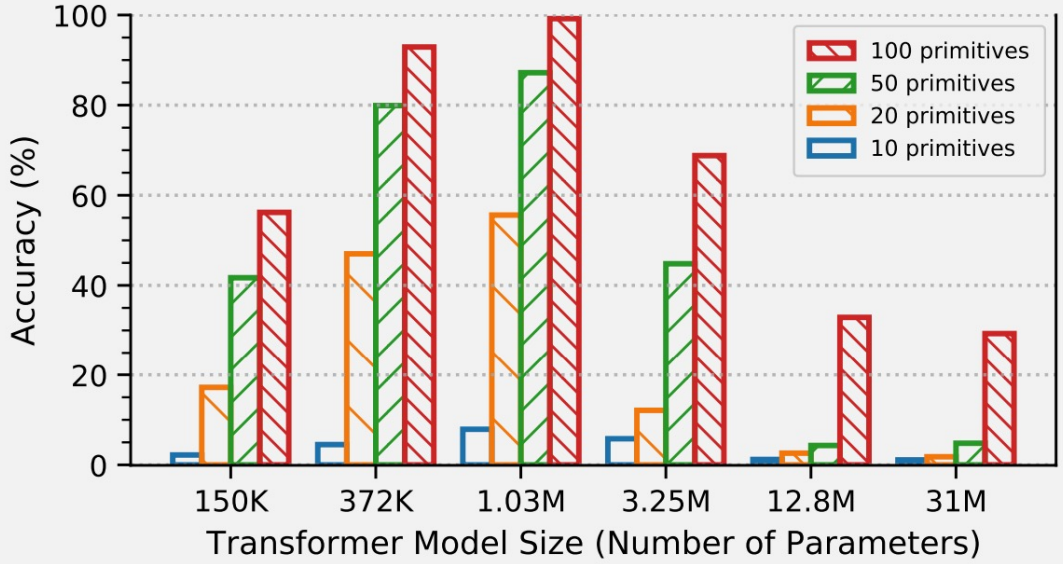


Figure 6: Measuring the generalization performance of a Transformer of varying capacity across increasing number of primitives in the SCAN training set.

Patel et al., Revisiting the Compositional Generalization Abilities of Neural Sequence Models, ACL 2022

Grokking

- **Grokking**: where a model initially overfits the training data, but after continuing training much longer, it suddenly starts to generalize well

- Grokking reveals that extended training can help models move from **memorization** to **true understanding** of the underlying patterns.

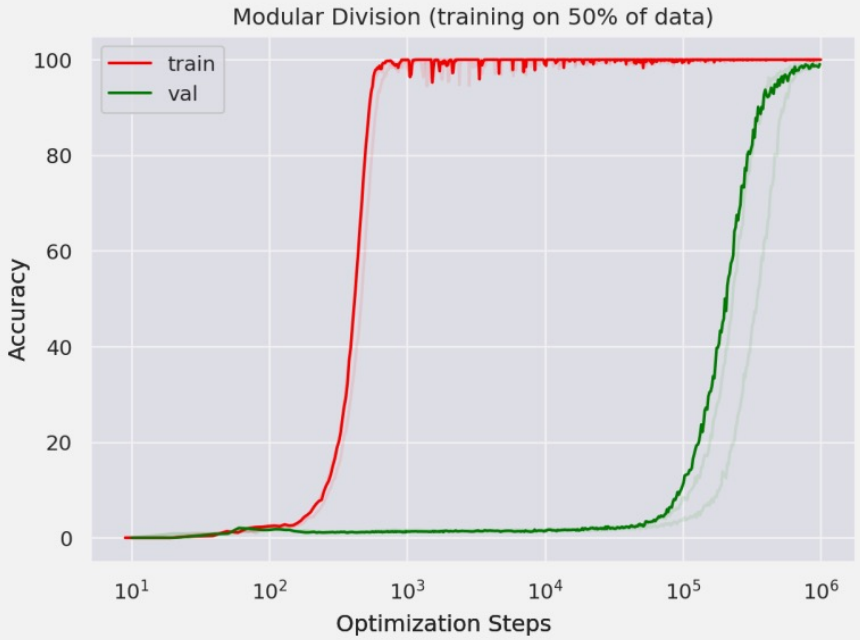
Grokking for Compositional Generalization

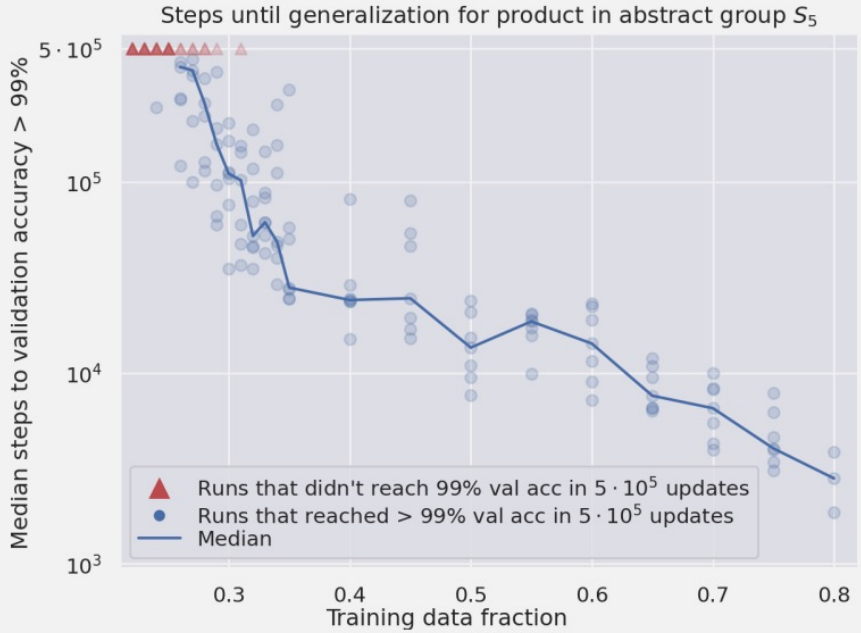
- NNs are capable of generalizing to the empty slots in a variety of binary operation tables.

- We show empirically that the amount of optimization required for generalization quickly increases as the dataset size decreases.

Power et al., Grokking: Generalization Beyond Overfitting on Small Algorithmic Datasets, 2022

Grokking for Compositional Generalization





★	a	b	c	d	e
a	a	d	?	c	d
b	c	d	d	a	c
c	?	e	d	b	d
d	a	?	?	b	c
e	b	b	c	?	a

Power et al., Grokking: Generalization Beyond Overfitting on Small Algorithmic Datasets, 2022

Grokked Transformers are Implicit Reasoners

- Transformers are capable of learning to implicitly reason over parametric knowledge

- It is only robustly acquired through extended training far beyond the point of overfitting, or grokking.

Wang et al., Grokked Transformers are Implicit Reasoners: A Mechanistic Journey to the Edge of Generalization, NeurIPS 2024

Two-Hop Composition

- E: entities that are used as subjects or objects
- Relations: The atomic facts as triplets, i.e., (subject, relation, object)
- **Composition**

$$\forall h, b, t \in \mathcal{E}, \forall r_1, r_2 \in \mathcal{R}, (h, r_1, b) \wedge (b, r_2, t) \implies (h, r_1, r_2, t)$$

- Goal: Can transformers **induce and apply latent rules** over knowledge in a **generalizable** way?

Wang et al., Grokked Transformers are Implicit Reasoners: A Mechanistic Journey to the Edge of Generalization, NeurIPS 2024

Training Data & ID/OOD Evaluation

- Data Generation Process:
  - Sample a set of **basic atomic facts**.
  - Use atomic facts + **latent rules** to deduce **inferred facts**.
- Data Splits: Atomic facts are partitioned disjointly into **atomicID** and **atomicOOD** (95%: 5%).

- Training Set: **all the atomic facts** and a uniformly random portion of the inferred facts deduced from atomicID (**train_inferredID**).
- Evaluation:
  - **ID Generalization:** Test on unseen inferred facts from **atomicID** (**test_inferredID**).
    - Measures if model learns latent rules correctly.
  - **OOD Generalization:** Test on inferred facts from **atomicOOD** (**test_inferredOOD**).
    - Measures systematicity: applying rules regardless of data distribution.

Wang et al., Grokked Transformers are Implicit Reasoners: A Mechanistic Journey to the Edge of Generalization, NeurIPS 2024

Composition vs. Comparison

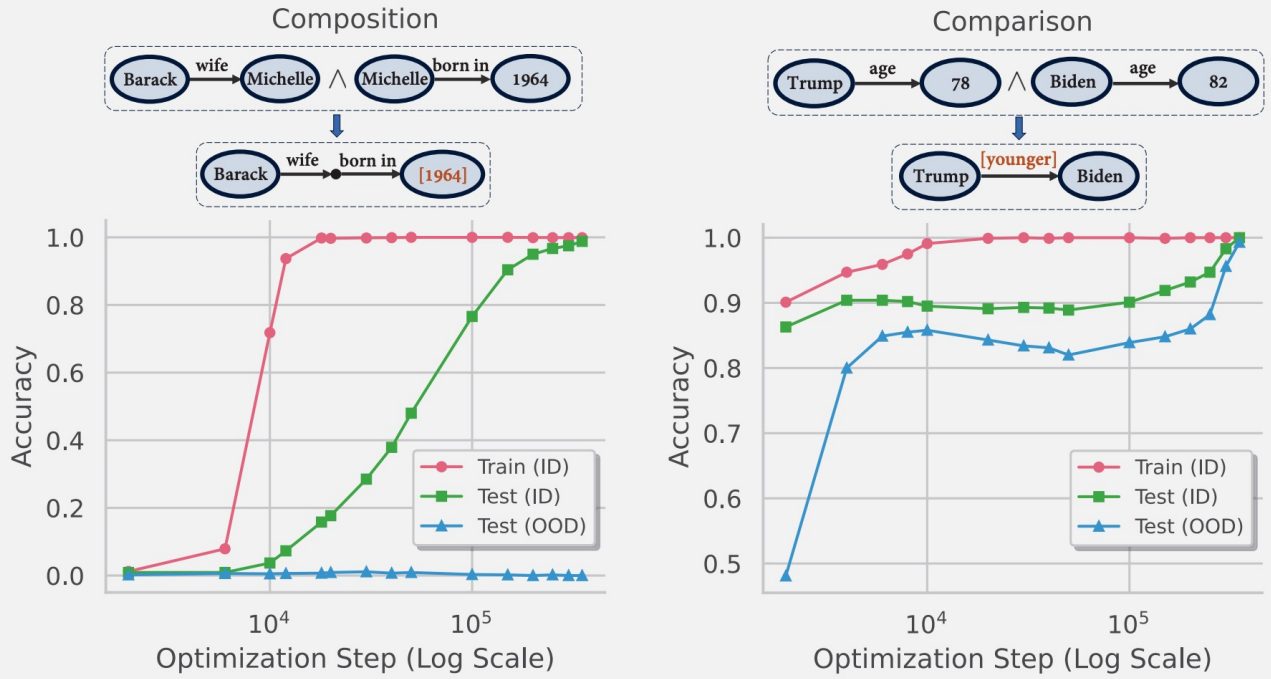
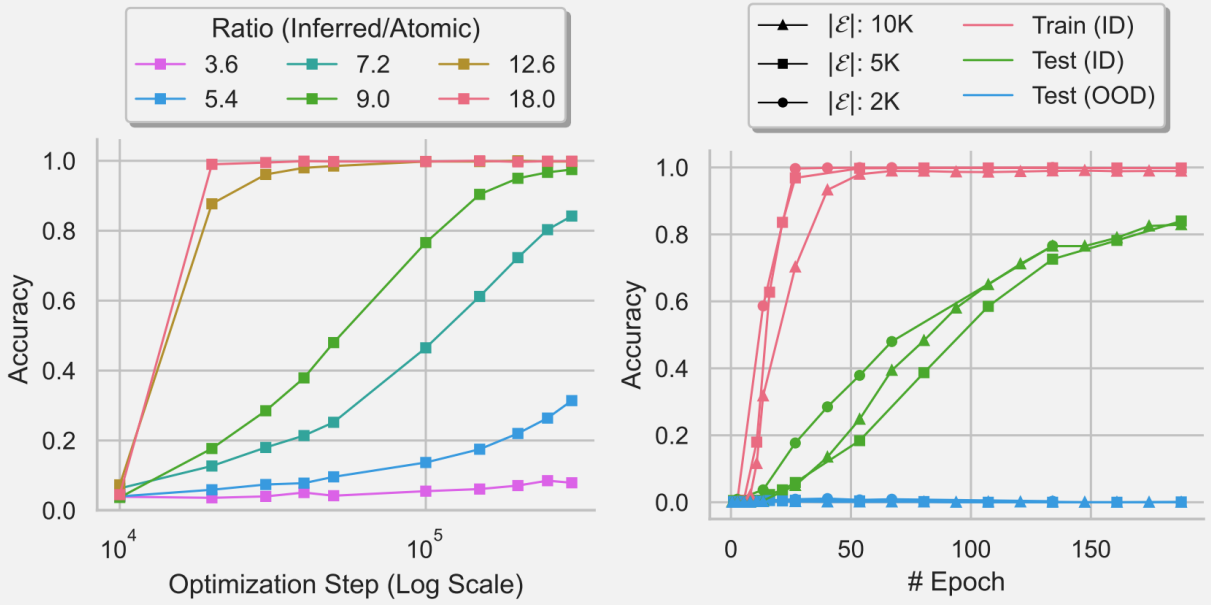


Figure 1: We find that transformers can learn to reason implicitly, but this skill is only robustly acquired through *grokking*, i.e., an extended period of training far beyond overfitting. Moreover, the transformer fails to *systematically* generalize for composition, yet succeeds for comparison. We conduct a mechanistic study into the model internals throughout grokking, which reveals distinct generalizing circuits across the two tasks (Figure 4, 5) that explains the variations in systematicity.

Wang et al., Grokked Transformers are Implicit Reasoners: A Mechanistic Journey to the Edge of Generalization, NeurIPS 2024

Inferred/Atomic Ratio ϕ and Generalization Speed

- Speed of improvement in ID generalization correlates with the ratio between inferred and atomic facts in training
- It depends little on the absolute size of the training data



(a) Effect of changing ratio ϕ ($|\mathcal{E}| = 2000$). (b) Effect of changing $|\mathcal{E}|$ ($\phi = 9.0$).

Wang et al., Grokked Transformers are Implicit Reasoners: A Mechanistic Journey to the Edge of Generalization, NeurIPS 2024

Composition - Delayed Generalization without Systematicity

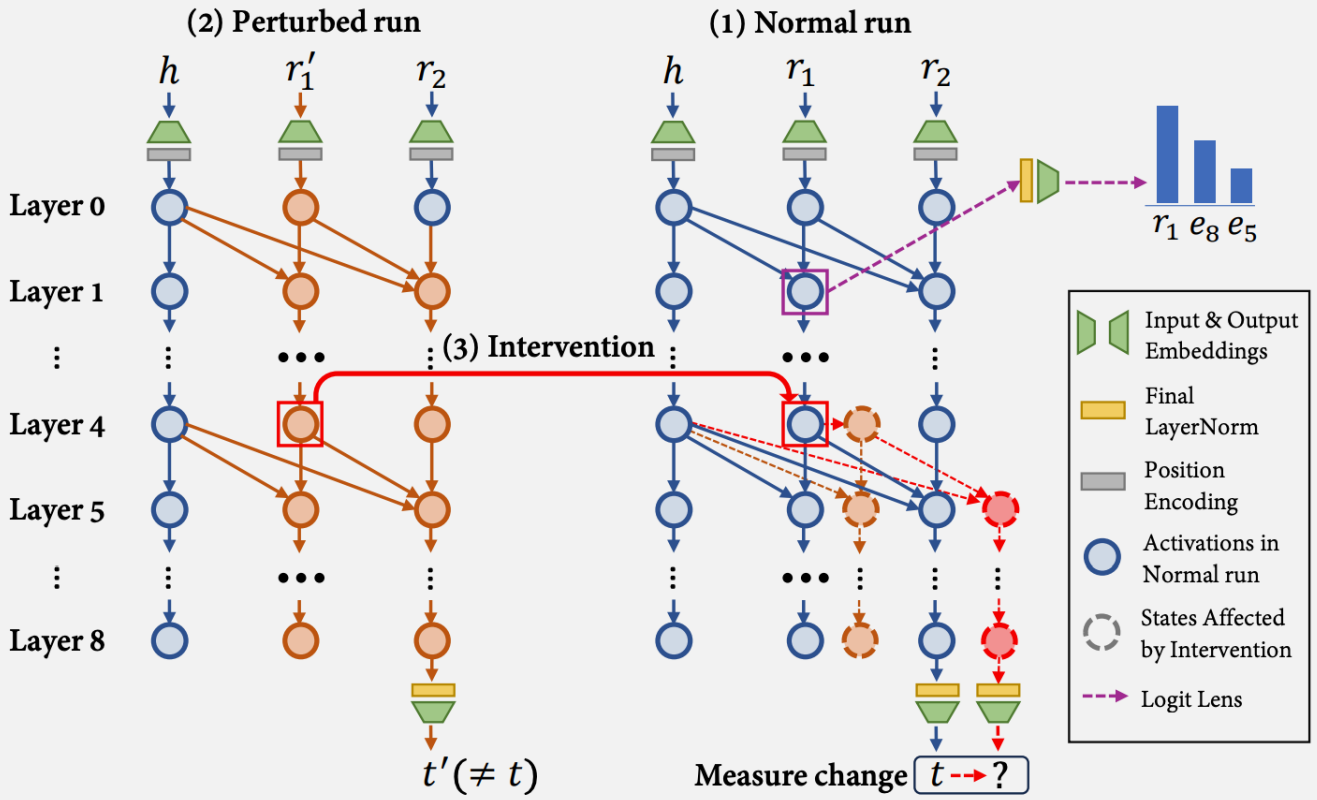
- Transformers are capable of acquiring the rule of composition through grokking (ID Generalization)
  - Critical data "distribution" (e.g., the inferred/atomic ratio ϕ), not size, may be the actual deciding factor

- Grokking is not observed in OOD generalization

- Open Question: What happens during grokking, why does it happen, and why do transformers struggle with OOD examples?

Wang et al., Grokked Transformers are Implicit Reasoners: A Mechanistic Journey to the Edge of Generalization, NeurIPS 2024

Analyzing the Inner Workings of the Model



Wang et al., Grokked Transformers are Implicit Reasoners: A Mechanistic Journey to the Edge of Generalization, NeurIPS 2024

Intervention

- Perturbation: for the hidden state of interest, the input token is replaced (e.g., $r_1 \rightarrow r_1'$) that leads to a different target prediction (e.g., $t \rightarrow t'$)

- **Causal tracing** to measure the strength of connections between states
  - intervene the state of interest by replacing its activation with the activation in the perturbed run
  - then run the remaining computations and measure if the target state is altered.
  - the ratio of such alterations (between 0 and 1) among characterizes the causal strength between the state of interest and the target.

Wang et al., Grokked Transformers are Implicit Reasoners: A Mechanistic Journey to the Edge of Generalization, NeurIPS 2024

Generalizing Circuit for Composition

The model is gradually forming the second hop in the upper layers (5-8) during grokking.

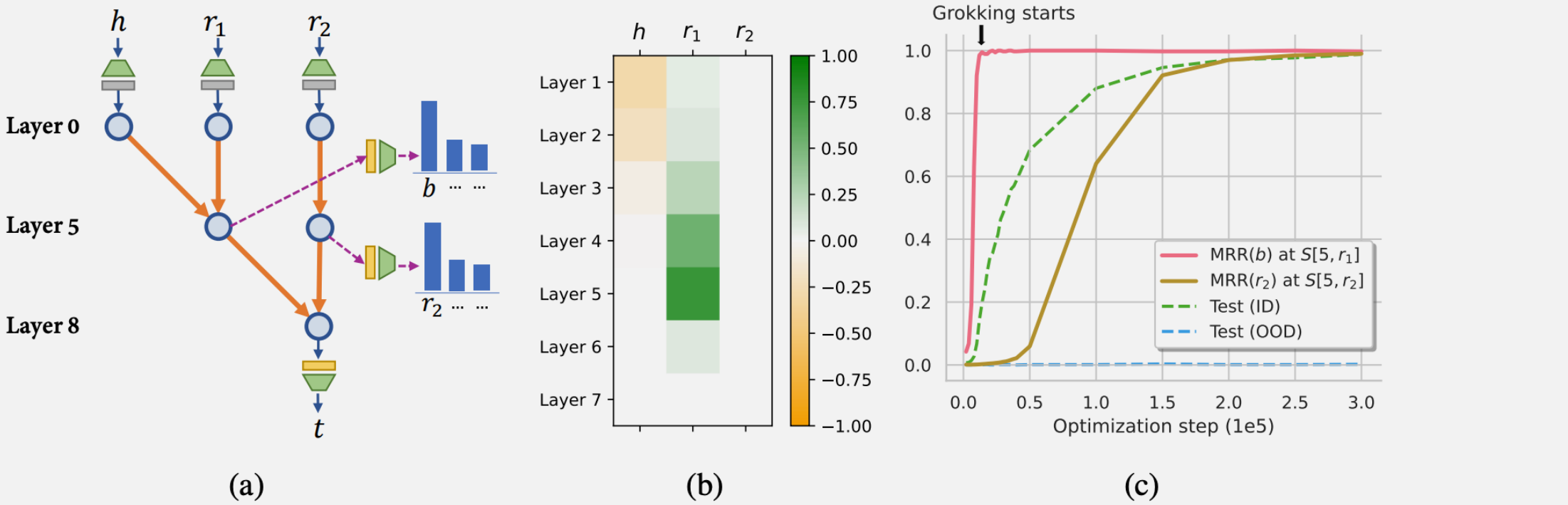


Figure 4: The (evolution of) generalizing circuit for composition. (a) The generalizing circuit. (b) The *change* in causal strengths during grokking, where the target is the prediction state. (c) Mean reciprocal rank (via logit lens) of the bridge entity b at $S[5, r_1]$ and second relation r_2 at $S[5, r_2]$.

Wang et al., Grokked Transformers are Implicit Reasoners: A Mechanistic Journey to the Edge of Generalization, NeurIPS 2024

What happens during grokking?

- The causal connection between S[5,r1] and the final prediction t, which is very weak before grokking, grows significantly during grokking

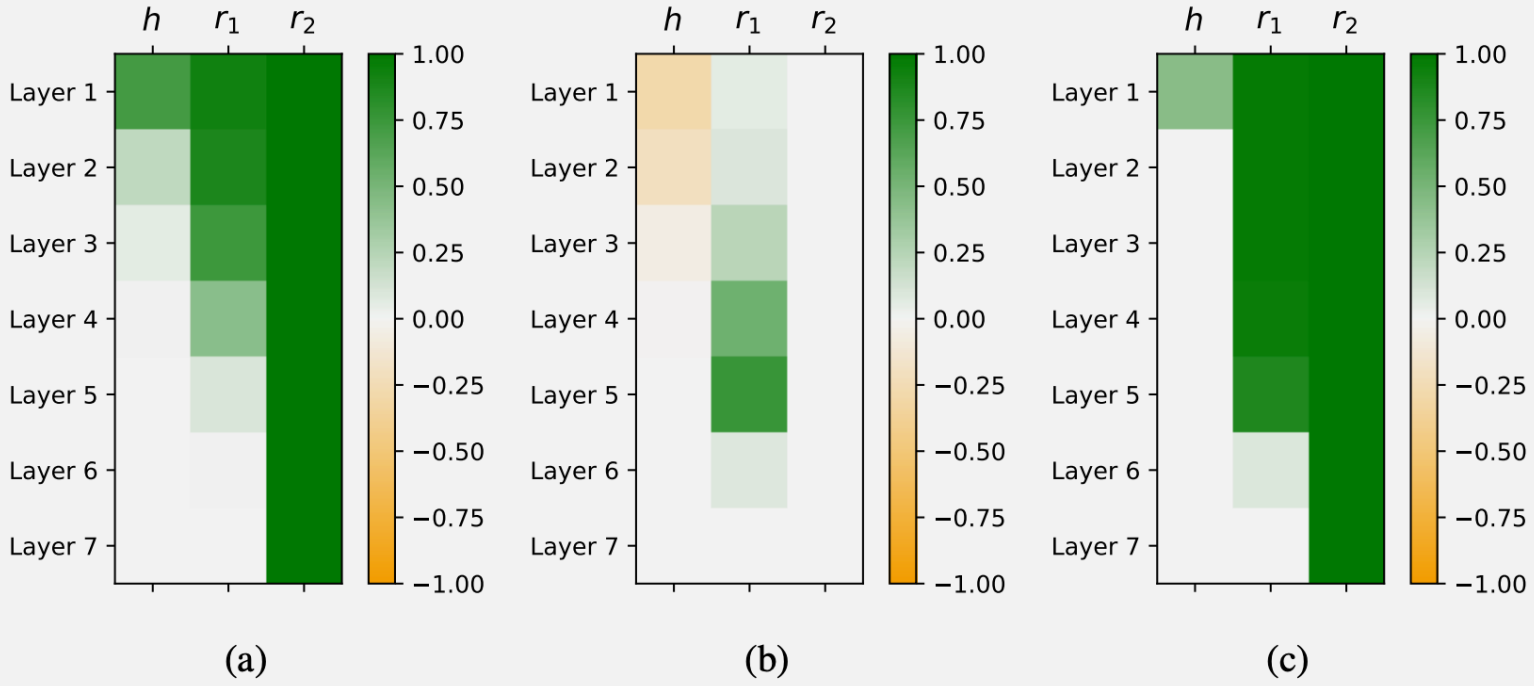


Figure 9: Causal strengths on composition task, with the final prediction $S[8, r_2]$ as the target. (a) Start of grokking. (b) Change during grokking. (c) End of grokking.

Wang et al., Grokked Transformers are Implicit Reasoners: A Mechanistic Journey to the Edge of Generalization, NeurIPS 2024

Why does grokking happen?

- C_{mem} stores both atomic facts and inferred facts in the weights.
- C_{gen} stores the atomic facts in the lower layers, and another copy of the atomic facts appearing as the second hop in the upper layers.
  - C_{gen} is relatively more efficient since it could fit with a lower complexity.

- As the inferred/atomic ratio ϕ increases, the relative efficiency of C_{gen} increases.
  - N_{mem} would increase rapidly while N_{gen} increases slowly (where the amount of facts C_{mem} and C_{gen} need to store are denoted as N_{mem} and N_{gen})
- In the long run, the model will be incentivized to transition from C_{mem} to C_{gen} due to implicit bias of the optimization and/or explicit regularization

Wang et al., Grokked Transformers are Implicit Reasoners: A Mechanistic Journey to the Edge of Generalization, NeurIPS 2024

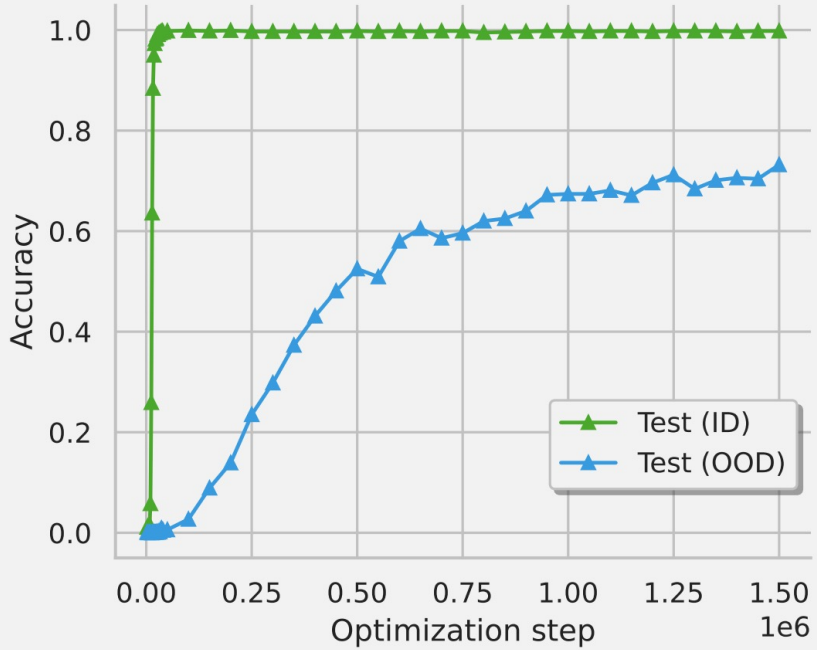
Why the model fails in the OOD setting?

- The OOD facts are only observed in the atomic form.
- The model does not have any incentive to store atomic facts in the upper layers that do not appear as the second hop during training.

Wang et al., Grokked Transformers are Implicit Reasoners: A Mechanistic Journey to the Edge of Generalization, NeurIPS 2024

Improving Transformer Design

- **Proper cross-layer memory-sharing mechanisms** such as memory-augmentation and explicit recurrence are needed to improve generalization.
- A variant of the parameter-sharing scheme in Universal Transformer (Dehghani et al., 2019) can improve OOD generalization in composition



Wang et al., Grokked Transformers are Implicit Reasoners: A Mechanistic Journey to the Edge of Generalization, NeurIPS 2024

Comparison Task

- The rules of comparison:

$$\forall e_1, e_2 \in \mathcal{E}, \forall a \in \mathcal{A}, \forall v_1, v_2 \in \mathcal{V}, (e_1, a, v_1) \wedge (e_2, a, v_2) \wedge v_1 < v_2 \implies (a, e_1, e_2, a_<),$$
$$\forall e_1, e_2 \in \mathcal{E}, \forall a \in \mathcal{A}, \forall v_1, v_2 \in \mathcal{V}, (e_1, a, v_1) \wedge (e_2, a, v_2) \wedge v_1 = v_2 \implies (a, e_1, e_2, a_=),$$
$$\forall e_1, e_2 \in \mathcal{E}, \forall a \in \mathcal{A}, \forall v_1, v_2 \in \mathcal{V}, (e_1, a, v_1) \wedge (e_2, a, v_2) \wedge v_1 > v_2 \implies (a, e_1, e_2, a_>).$$

Wang et al., Grokked Transformers are Implicit Reasoners: A Mechanistic Journey to the Edge of Generalization, NeurIPS 2024

Generalizing Circuit for Comparison

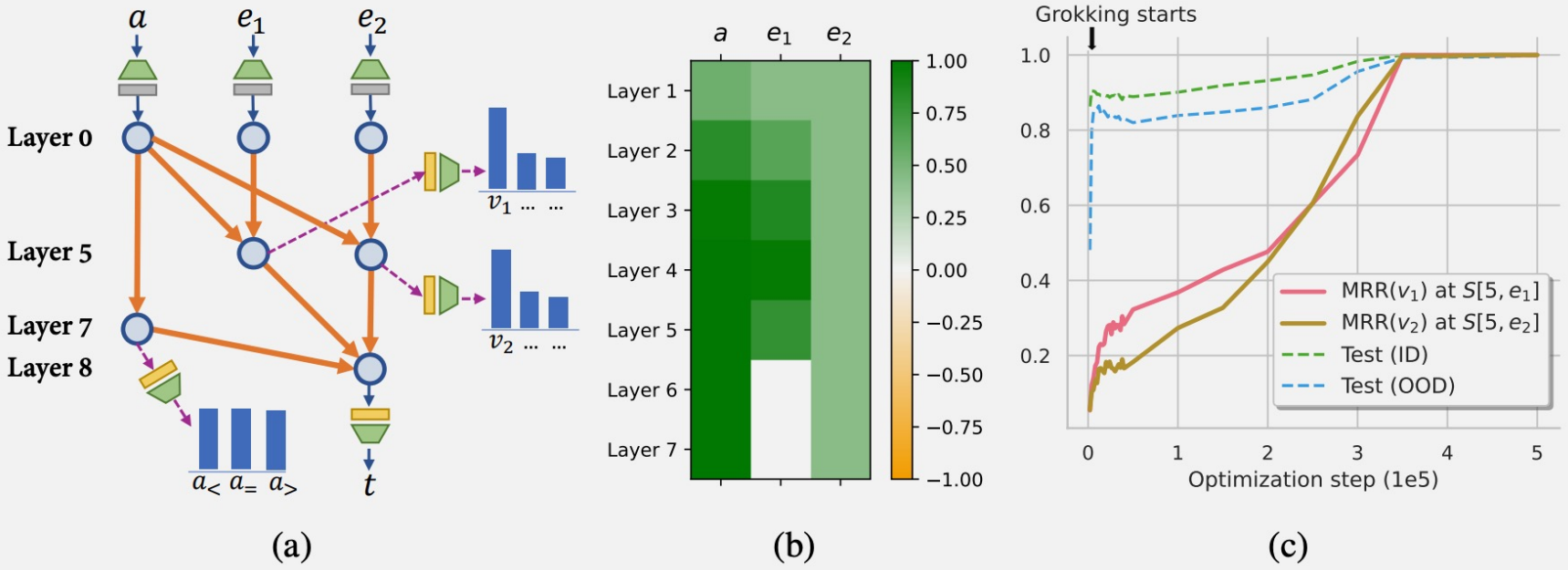


Figure 5: The (evolution of) generalizing circuit for comparison. (a) The generalizing circuit. (b) The *change* in causal strengths during grokking, where the target is the prediction state. (c) Mean reciprocal rank (via logit lens) of the two attribute values (v_1, v_2) at $S[5, e_1]$ and $S[5, e_2]$.

Wang et al., Grokked Transformers are Implicit Reasoners: A Mechanistic Journey to the Edge of Generalization, NeurIPS 2024

Systematic Generalization in Comparison

- The comparison task emits a “parallel circuit” that is learned by the transformer during grokking
  - allows atomic facts to be stored and retrieved in the same region and enables systematicity to happen

Wang et al., Grokked Transformers are Implicit Reasoners: A Mechanistic Journey to the Edge of Generalization, NeurIPS 2024

Composition vs. Comparision in Transformers

- There is a difference in the acquired generalization across the two tasks
- Transformer’s ability to learn parallel solutions to seemingly sequential problems

Wang et al., Grokked Transformers are Implicit Reasoners: A Mechanistic Journey to the Edge of Generalization, NeurIPS 2024

Compositional problems and LLMs

- Compositional problems require strict multi-hop reasoning
  - Under what conditions do transformers succeed, fail, and why?
  - What types of errors do they make?
  - Can transformers uncover implicit problem-solving rules or be taught to follow reasoning paths?

Dziri et al., Faith and Fate: Limits of Transformers on Compositionality, NeurIPS 2023

Hypotheses for Limits of Transformers on Compositionality

- First, transformers solve compositional tasks by reducing multi-step compositional reasoning into linearized path matching.

- Second, due to error propagation, transformers may have inherent limitations on solving novel high-complexity compositional tasks
  - Errors in the early stages of the computational process can lead to substantial compounding errors in subsequent steps

Dziri et al., Faith and Fate: Limits of Transformers on Compositionality, NeurIPS 2023

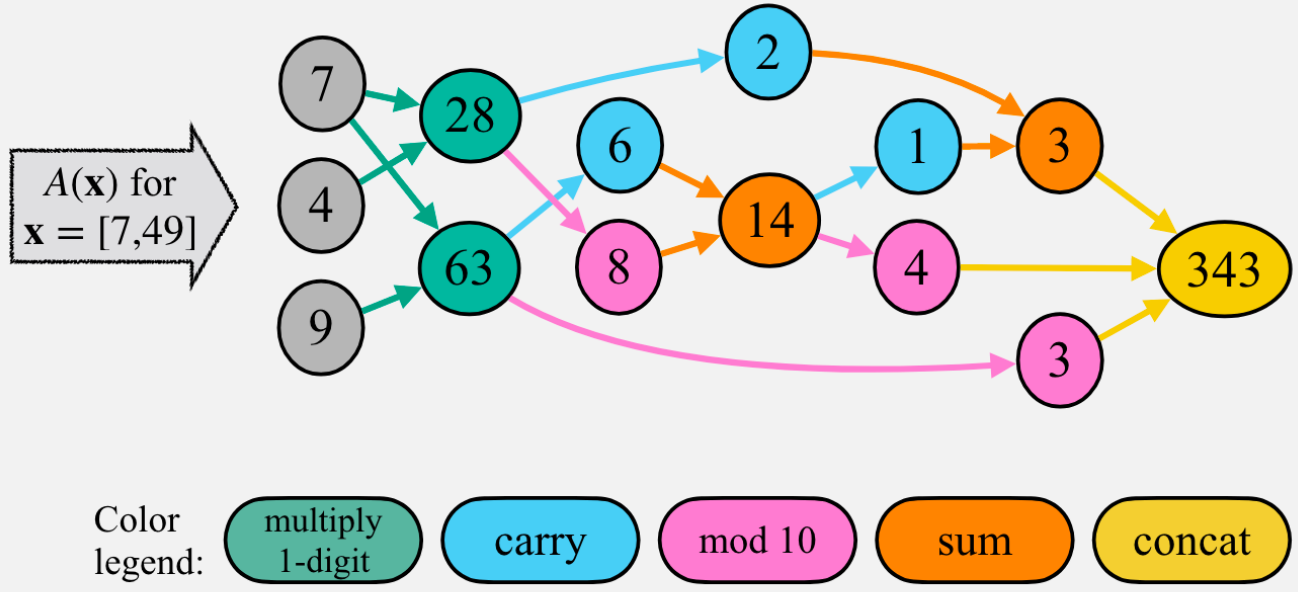
Formulating Algorithms by Computational Graph

- It formulates compositional tasks as computation graphs.
- Linearize GA(x): Since we only consider autoregressive models, this linearization must also be a topological ordering.

```
function multiply (x[1..p], y[1..q]):
  // multiply x for each y[i]
  for i = q to 1
    carry = 0
    for j = p to 1
      t = x[j] * y[i]
      t += carry
      carry = t // 10
      digits[j] = t mod 10
    summands[i] = digits

  // add partial results (computation not shown)
  product = ∑_{i=1}^{q} summands[q+1-i] · 10^{i-1}
  return product
```

$A(\mathbf{x})$



Dziri et al., Faith and Fate: Limits of Transformers on Compositionality, NeurIPS 2023

Quantifying Compositional Complexity by Graph Metrics

- **Reasoning depth**: the largest layer number in the graph.
  - a proxy for the max level of multi-hop reasoning required to solve the task.
- **Reasoning width**: the max number of variables required to maintain in parallel during the computation.
- **Average parallelism**: the ratio between |V| and reasoning depth
  - This aims to compute the average width in computation through the graph, and not just in its mode.

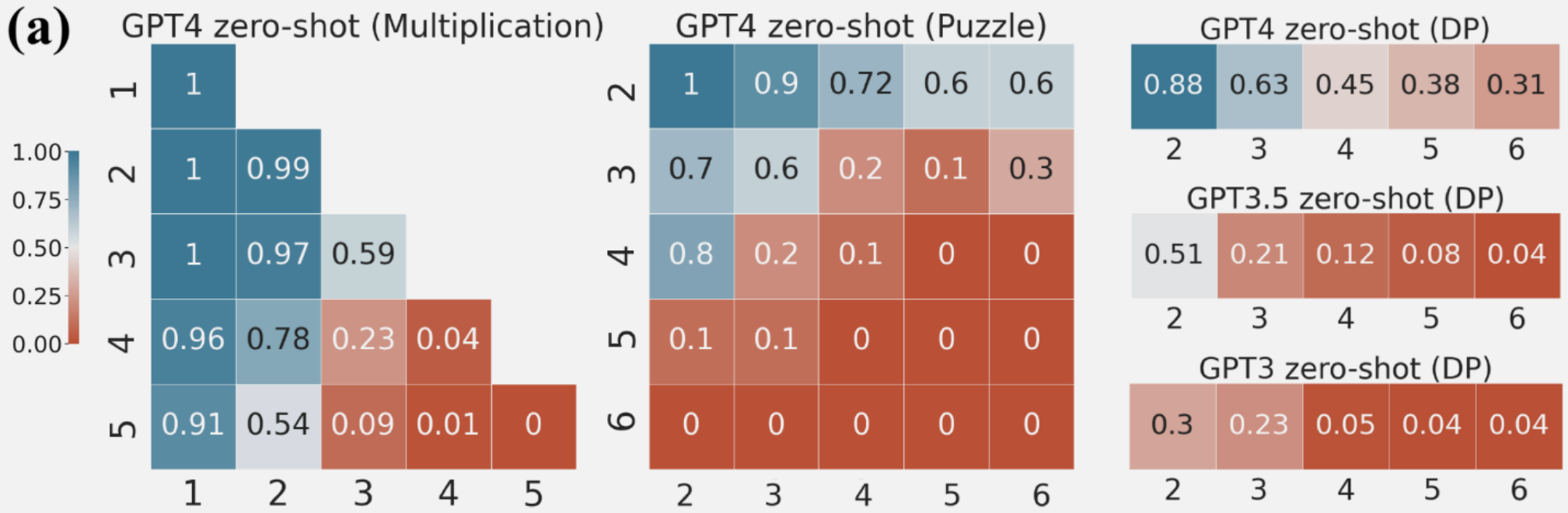
Dziri et al., Faith and Fate: Limits of Transformers on Compositionality, NeurIPS 2023

Compositional Tasks

- Three representative compositional tasks
  - multi-digit multiplication
  - logic grid puzzles
  - a classic dynamic programming problem.

Dziri et al., Faith and Fate: Limits of Transformers on Compositionality, NeurIPS 2023

Limits of Transformers in Zero-Shot Settings



Dziri et al., Faith and Fate: Limits of Transformers on Compositionality, NeurIPS 2023

Limits of Transformers with Question-Answer Training

- separately finetune GPT3 models on:
  - ~1.8M multiplication pairs
  - ~142K DP pairs
  - ~41K puzzle pairs

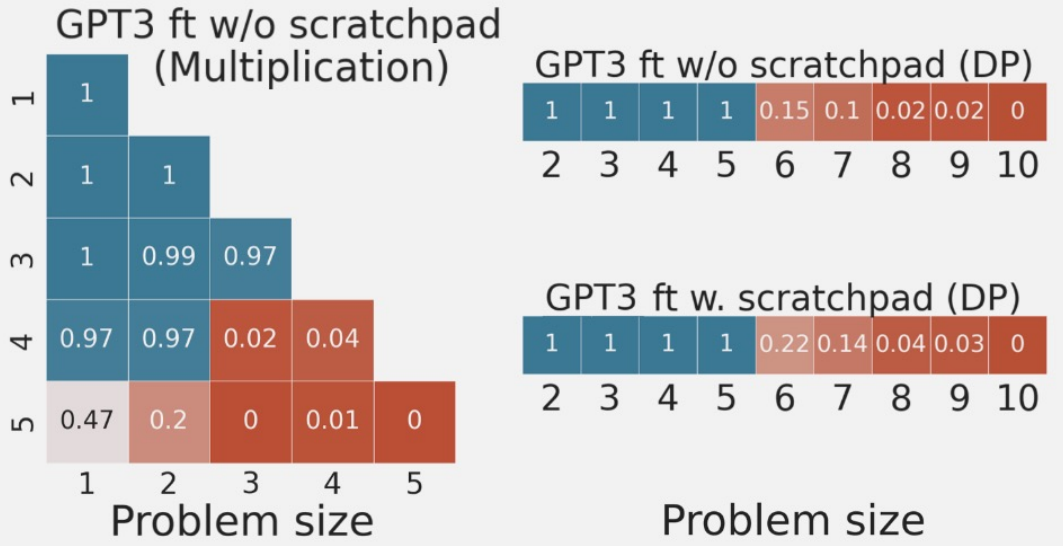
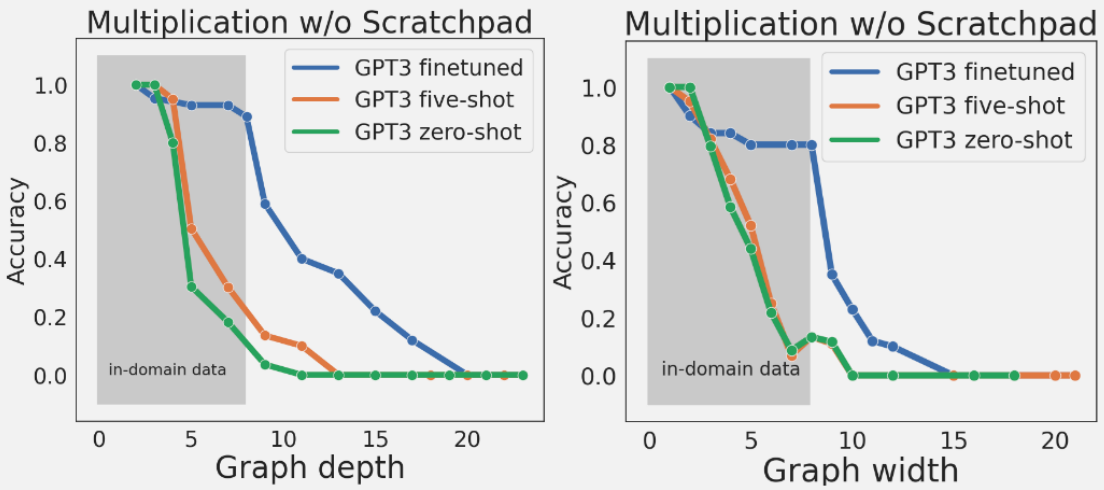


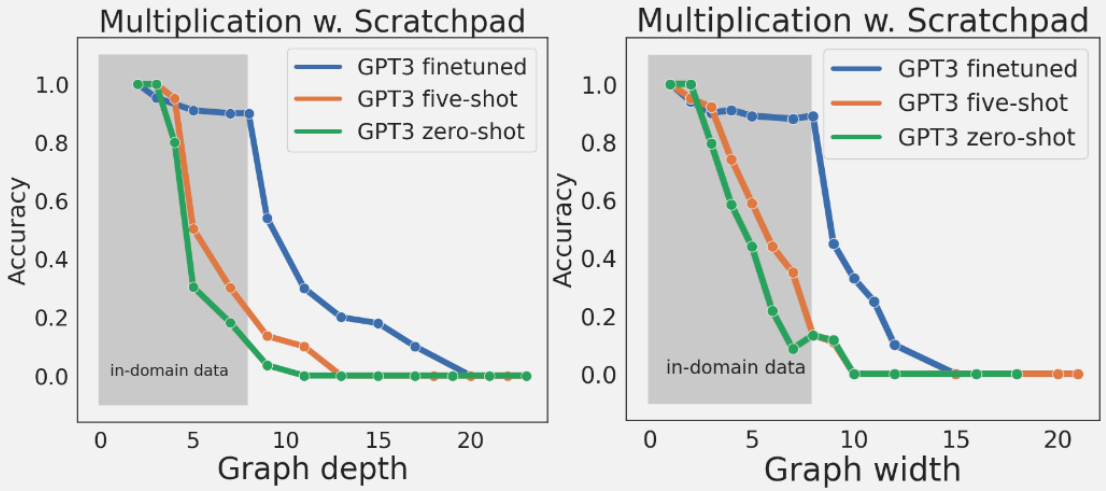
Figure 3: GPT3 finetuned exhaustively on task-specific data up to a certain problem size. The **blue** region represents the in-distribution examples and the **red** region refers to OOD examples. The same trend is observed for the puzzle task (See §B.2)

Dziri et al., Faith and Fate: Limits of Transformers on Compositionality, NeurIPS 2023

Limits of Transformers with Question-Answer Training



(a) Results on **question-answer** pairs.



(b) Results on **question-scratchpad** pairs.

Dziri et al., Faith and Fate: Limits of Transformers on Compositionality, NeurIPS 2023

Relative Information Gain & Surface Patterns

- Relative Information Gain
  - Task T is represented as a distribution $(X_{1:n}, Y_{1:m})$ and measure the amount of (normalized) information gained about an Y_j and $X \subset \{X_1, \ldots, X_n\}$:

$$\text{RelativeIG}(Y_j, X) = \frac{H(Y_j) - H(Y_j|X)}{H(Y_j)} \in [0, 1]$$

- It is leveraged to predict patterns that models are likely to learn based on the underlying task distribution

Dziri et al., Faith and Fate: Limits of Transformers on Compositionality, NeurIPS 2023

Surface Form Clues

- An output element may rely heavily on a single or small set of features
- Transformers can exploit these correlations during training.

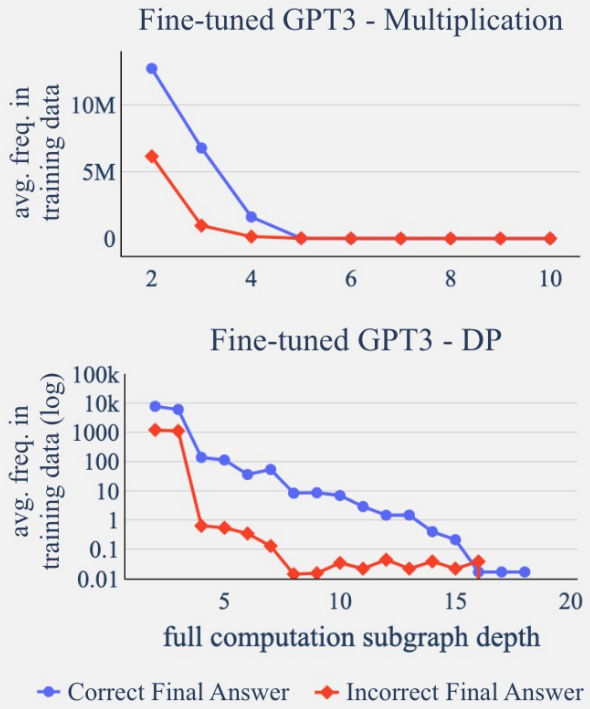
- At test time, the model may directly map inputs to outputs w/o true reasoning.
  - This behavior can bypass multi-hop reasoning
  - Leading to correct predictions, but a false impression of compositional reasoning.

		Relative Information Gain			
Input variable	Output variable	2x2	3x3	4x4	5x5
x_n	z_{2n}	0.223	0.223	0.223	0.223
y_n	z_{2n}	0.223	0.223	0.223	0.223
x_1	z_1	0.198	0.199	0.199	0.199
y_1	z_1	0.198	0.199	0.199	0.199
$x_n\ y_n$	z_{2n}	1.000	1.000	1.000	1.000
$x_{n-1}\ x_n$	z_{2n}	0.223	0.223	0.223	0.223
$y_{n-1}\ y_n$	z_{2n}	0.223	0.223	0.223	0.223
$x_n\ y_n$	z_{2n-1}	0.110	0.101	0.101	0.101
$y_{n-1}\ y_n$	z_{2n-1}	0.032	0.036	0.036	0.036
$x_{n-1}\ x_n$	z_{2n-1}	0.032	0.036	0.036	0.036
$x_{n-1}\ y_{n-1}$	z_{2n-1}	0.018	0.025	0.025	0.025
$x_1\ y_1$	z_2	0.099	0.088	0.088	0.088
$x_2\ y_2$	z_2	0.025	0.016	0.016	0.016
$x_1\ y_1$	z_1	0.788	0.792	0.793	0.793
$y_1\ y_2$	z_1	0.213	0.211	0.211	0.211
$x_1\ x_2$	z_1	0.213	0.211	0.211	0.211

Dziri et al., Faith and Fate: Limits of Transformers on Compositionality, NeurIPS 2023

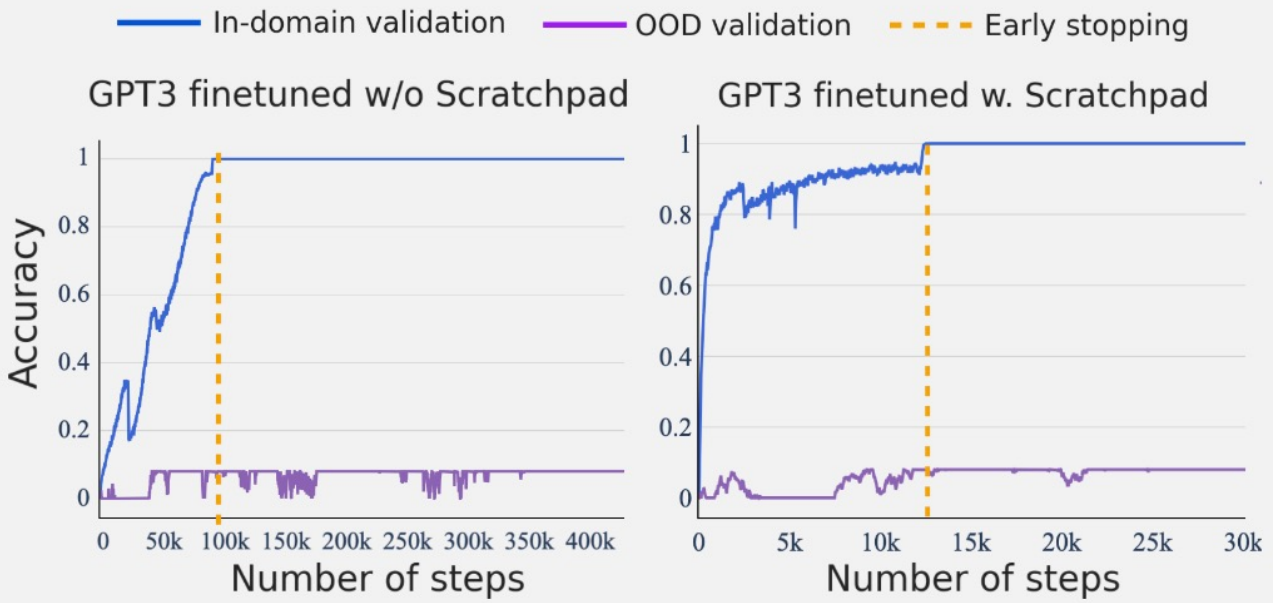
Linearized Subgraph Matching instead of Multi-Step Compositional Reasoning

- Multi-step compositional reasoning is reduced into linearized subgraph matching
- Full computation subgraphs appear significantly more frequently in the training data for correctly predicted test examples than for incorrectly predicted ones



Dziri et al., Faith and Fate: Limits of Transformers on Compositionality, NeurIPS 2023

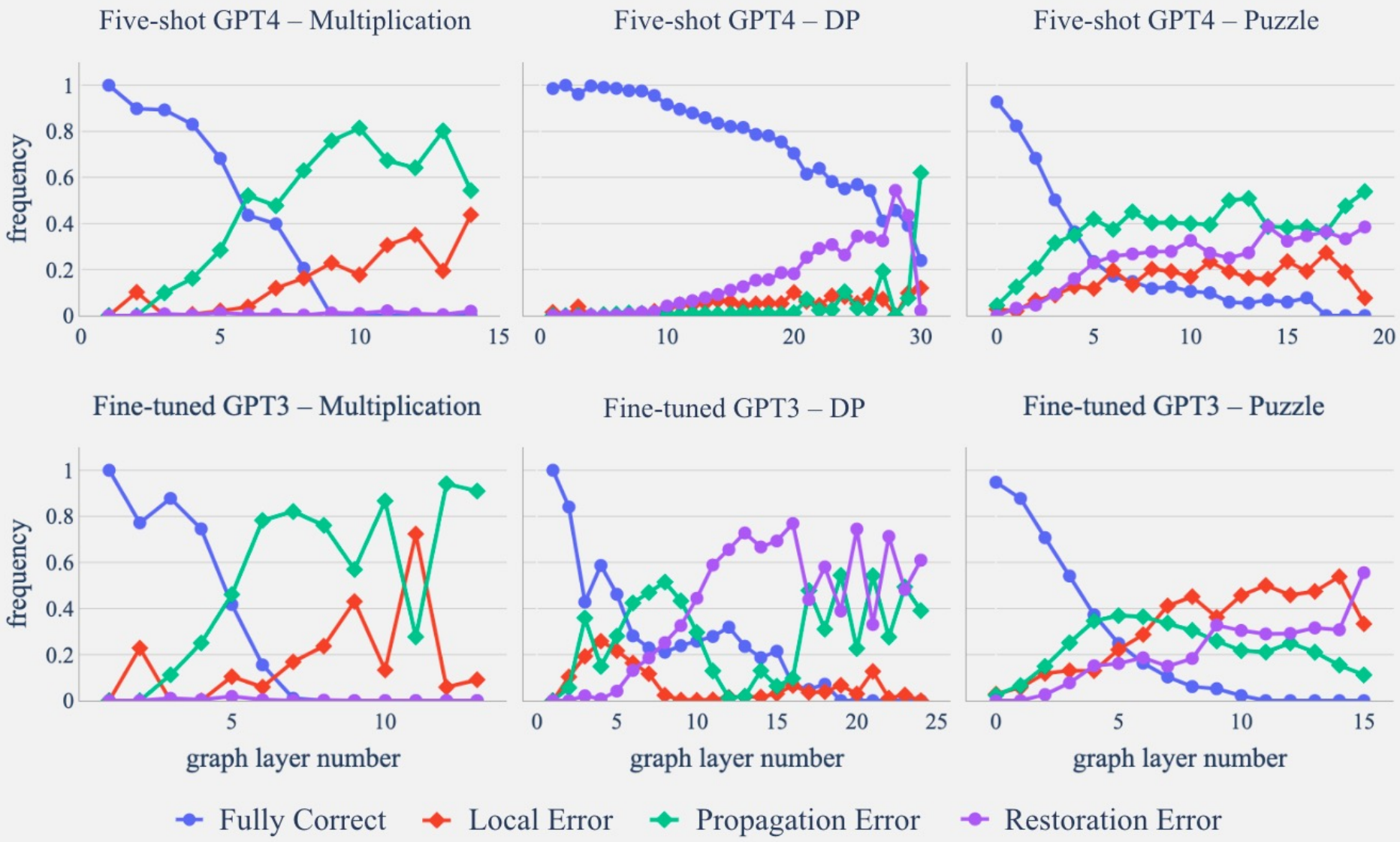
Limits of Transformers with Grokking



Dziri et al., Faith and Fate: Limits of Transformers on Compositionality, NeurIPS 2023

Types of Errors

- v is fully correct if v and its ancestors have correct values and are derived from correct computations.
- v has a local error if its parent nodes have correct values but v is derived from an incorrect computation (i.e., a one-hop reasoning error);
- v has a propagation error if v is derived from a correct computation but some of its parent nodes have incorrect values;
- v has a restoration error if it has a correct value but is derived from an incorrect computation.



Dziri et al., Faith and Fate: Limits of Transformers on Compositionality, NeurIPS 2023

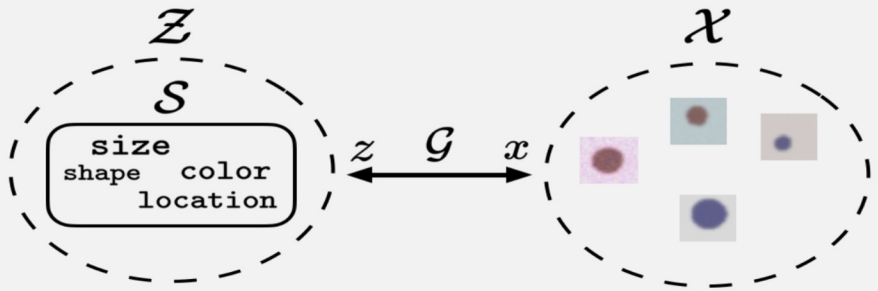
Summary of Results

- Near-perfect ID generalization under low compositional complexity, but fails drastically on OOD instances
  - This substantial gap suggests that systematic problem-solving capabilities do not emerge from maximum likelihood training
- While models can memorize single-step operations, they fail to compose them into correct reasoning paths
  - mostly make predictions based on shallow, rote learning rather than a holistic task understanding
- Transformers reduce multi-step compositional reasoning into linearized subgraph matching, without necessarily developing systematic problem-solving skills.

Dziri et al., Faith and Fate: Limits of Transformers on Compositionality, NeurIPS 2023

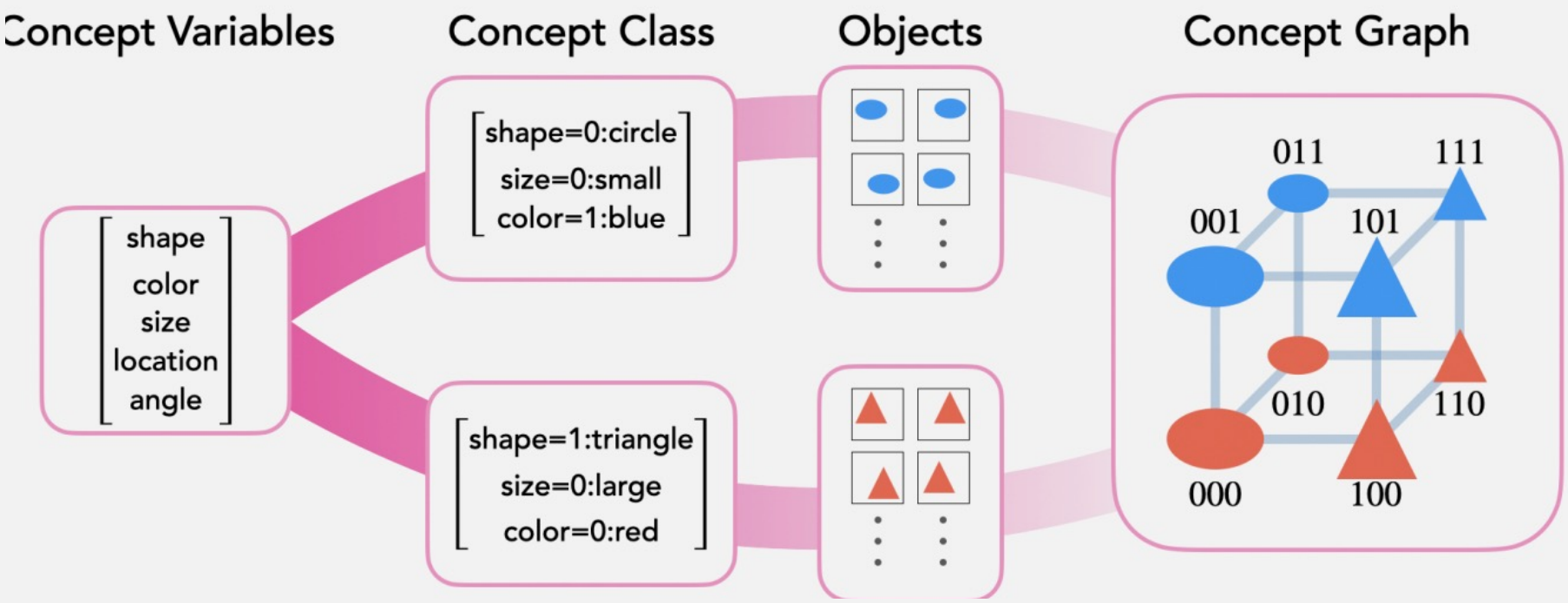
Concept Space

Definition 1. ***(Concept Space.)*** *Consider an invertible data-generating process $\mathcal{G} : \mathcal{Z} \to \mathcal{X}$ that samples vectors $z \sim P(\mathcal{Z})$ from a vector space $\mathcal{Z} \subset \mathbb{R}^d$ and maps them to the observation space $\mathcal{X} \in \mathbb{R}^n$. We assume the sampling prior is factorizable, i.e., $P(z \in \mathcal{Z}) = \Pi_{i=1}^{d} P(z_i)$, and individual dimensions of $\mathcal{Z}$ correspond to semantically meaningful concepts. Then, a concept space $\mathcal{S}$ is defined as the multidimensional space composed of all possible concept vectors z, i.e., $\mathcal{S} \coloneqq \{z \mid z \sim P(\mathcal{Z})\}$*



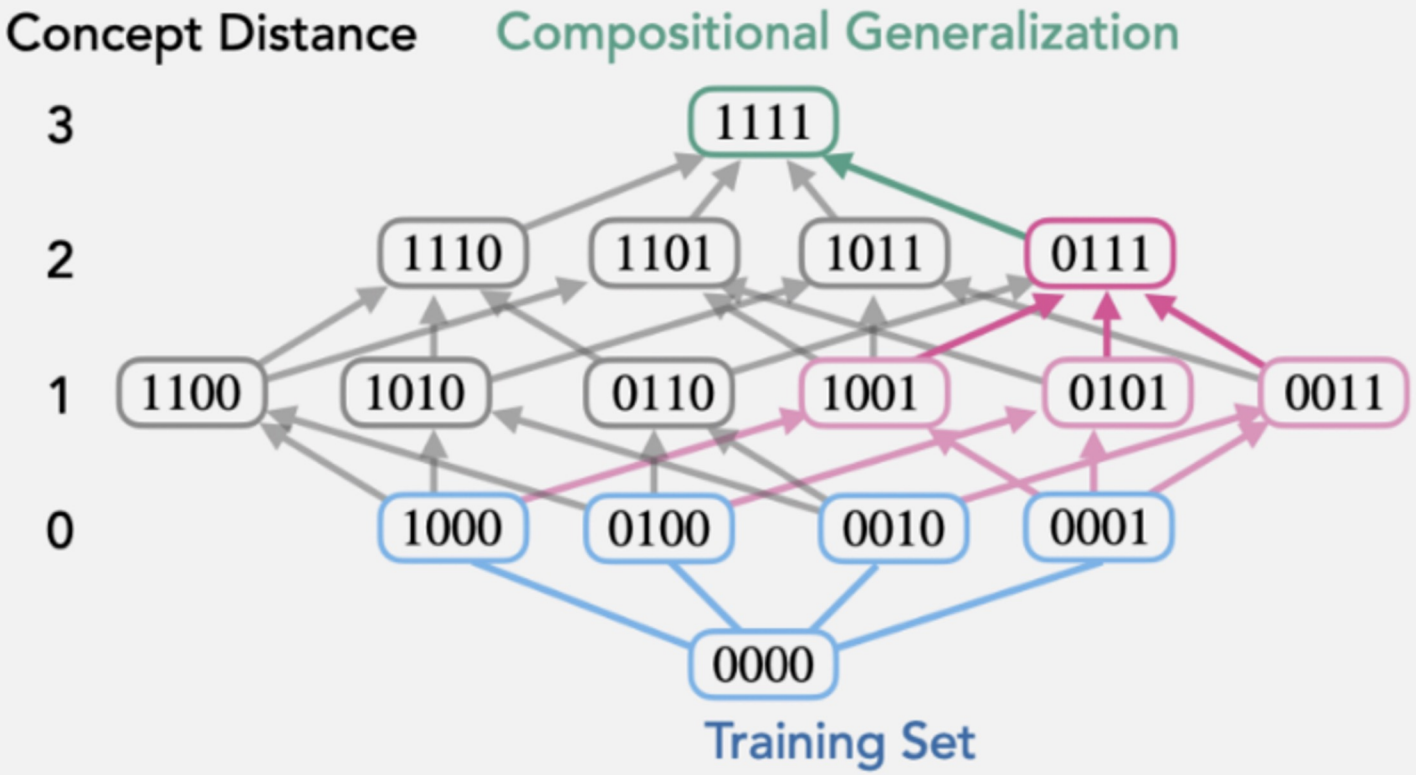
Park et al., Emergence of Hidden Capabilities: Exploring Learning Dynamics in Concept Space, NeurIPS 2024

Concept Graph



Okawa et al., Compositional Abilities Emerge Multiplicatively: Exploring Diffusion Models on a Synthetic Task, NeurIPS 2023

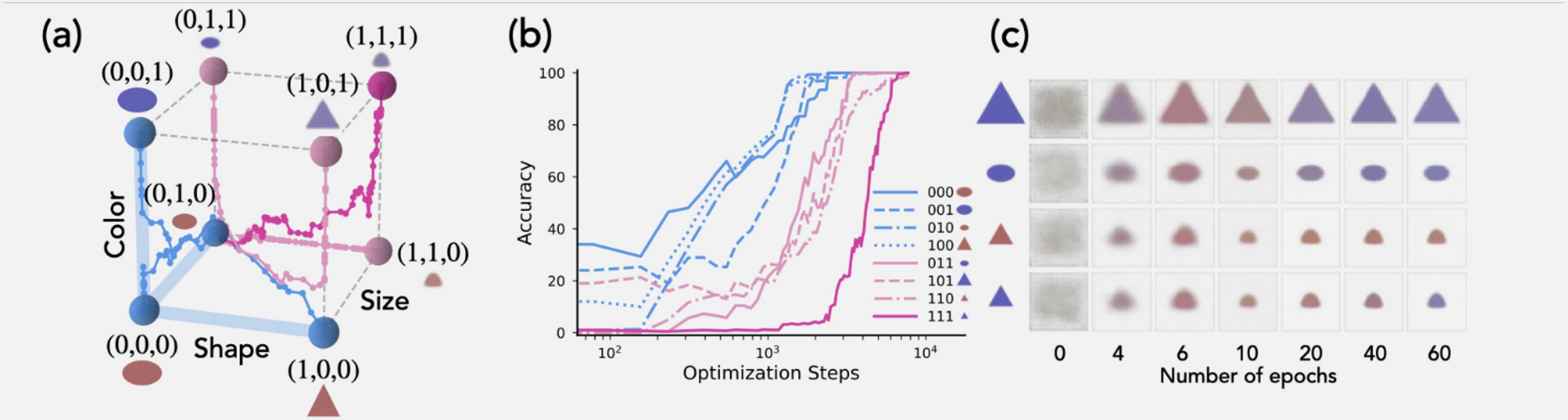
Concept Distance



Okawa et al., Compositional Abilities Emerge Multiplicatively: Exploring Diffusion Models on a Synthetic Task, NeurIPS 2023

Concept Distance & Compositional

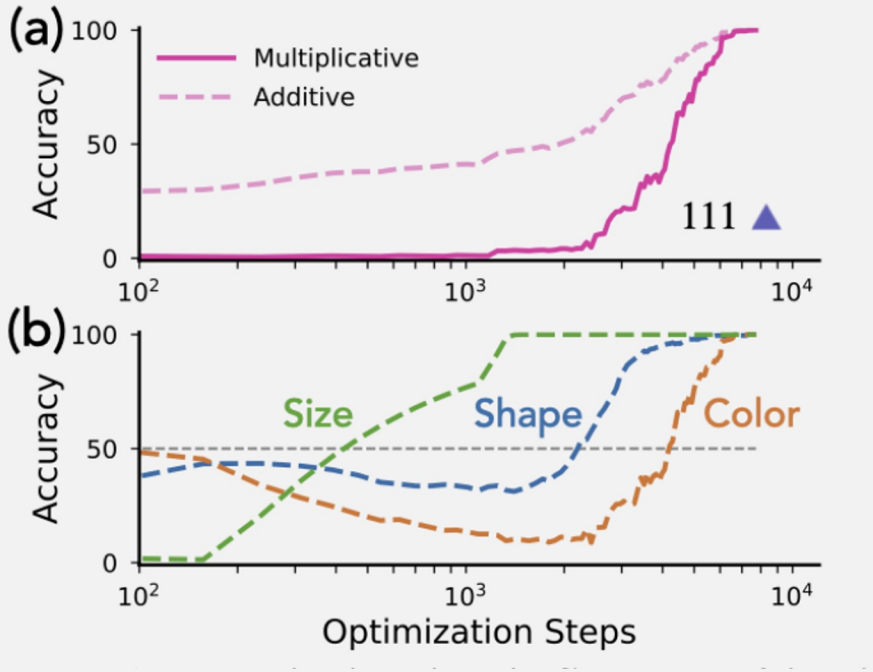
- Concept distance from the training set governs the order in which compositional capabilities emerge



Okawa et al., Compositional Abilities Emerge Multiplicatively: Exploring Diffusion Models on a Synthetic Task, NeurIPS 2023

Multiplicative influence of individual capabilities

- Multiplicative influence of individual capabilities elicits an “emergence” of compositionality



Okawa et al., Compositional Abilities Emerge Multiplicatively: Exploring Diffusion Models on a Synthetic Task, NeurIPS 2023

Hypothesis: Compositional Abilities Emerge Multiplicatively

- Hypothesize: the nonlinear increase in capability as size and computational power scale up is partially driven by the task's compositionality.

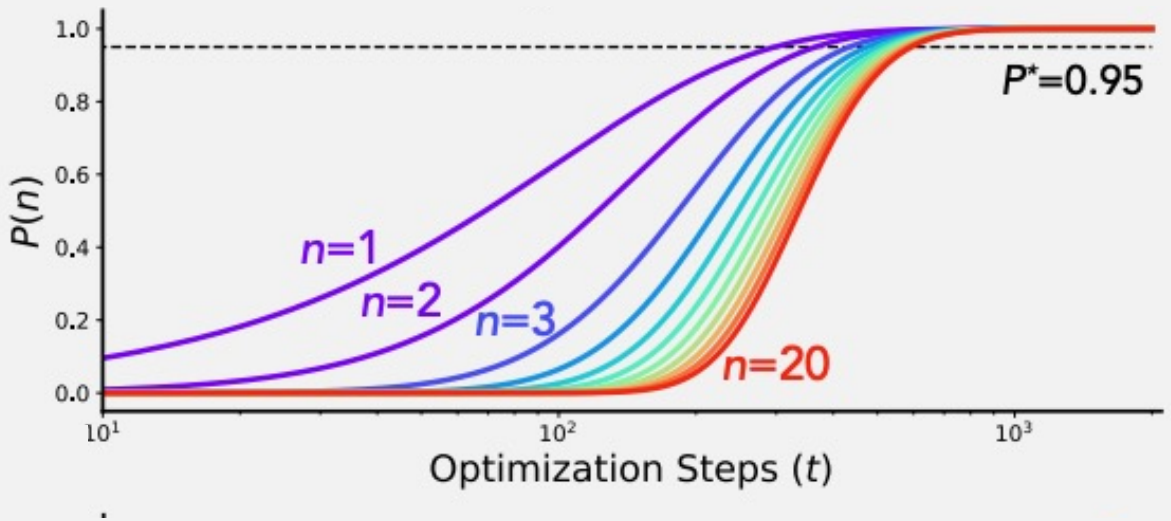
- Models must learn all required concepts, but compositional generalization is hindered by the multiplicative impact of learning progress on each concept.

- This results in a rather sudden emergence of capabilities to produce or reason about out-of-distribution data.

Okawa et al., Compositional Abilities Emerge Multiplicatively: Exploring Diffusion Models on a Synthetic Task, NeurIPS 2023

Sudden Emergence of Capabilities

- n: no of atomic abilities
- p: the probability of being learned an ability in a given time step
- The probability of an ability will be learned in t steps: $1 - (1 - p)^t$
- The probability of learning a compositioal capability by time t: $P(n) = (1 - (1 - p)^t)^n$.



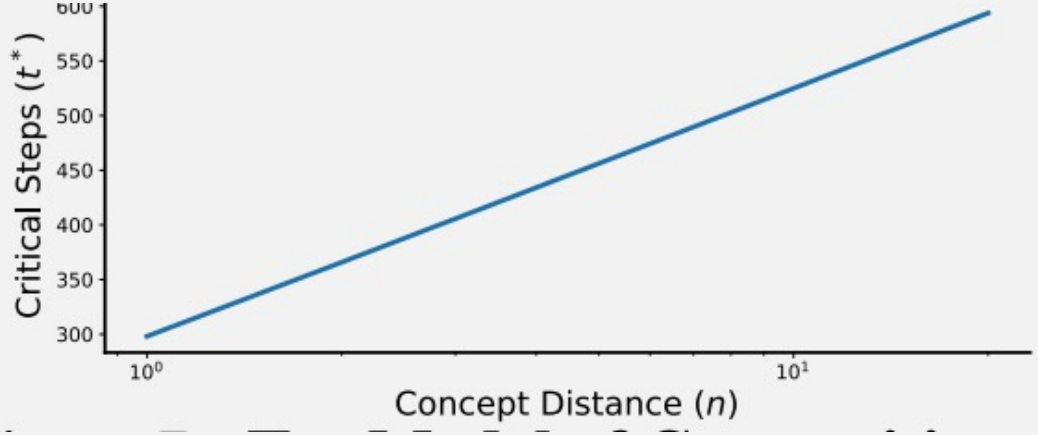
Okawa et al., Compositional Abilities Emerge Multiplicatively: Exploring Diffusion Models on a Synthetic Task, NeurIPS 2023

Logarithmic Progress of Compositional Tasks

- P^*: a threshold probability at which one claims a capability has been learned, for a degree of composition n
- The critical time t^* at which a compositional capability is learned can be found as:

$$t^* = \left\lceil \frac{\log 1 - (P^*)^{1/n}}{\log 1 - P} \right\rceil$$

- The progress on increasingly more compositional tasks is logarithmic w.r.t. the number of atomic concepts being composed (aka concept distance).

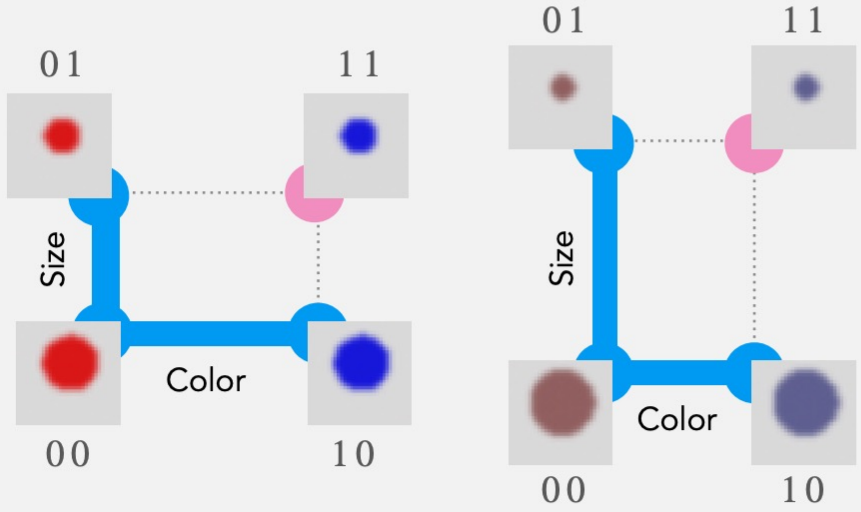


Okawa et al., Compositional Abilities Emerge Multiplicatively: Exploring Diffusion Models on a Synthetic Task, NeurIPS 2023

Concept Signal

Definition 3. ***(Concept Signal.)*** *The concept signal σ_i for a concept z_i measures the sensitivity of the data-generating process to change in the value of a concept variable, i.e., $\sigma_i := \left|\partial \mathcal{G}(z) / \partial z_i\right|$.*

- Concept spaces with different concept values see different concept signal



Park et al., Emergence of Hidden Capabilities: Exploring Learning Dynamics in Concept Space, NeurIPS 2024

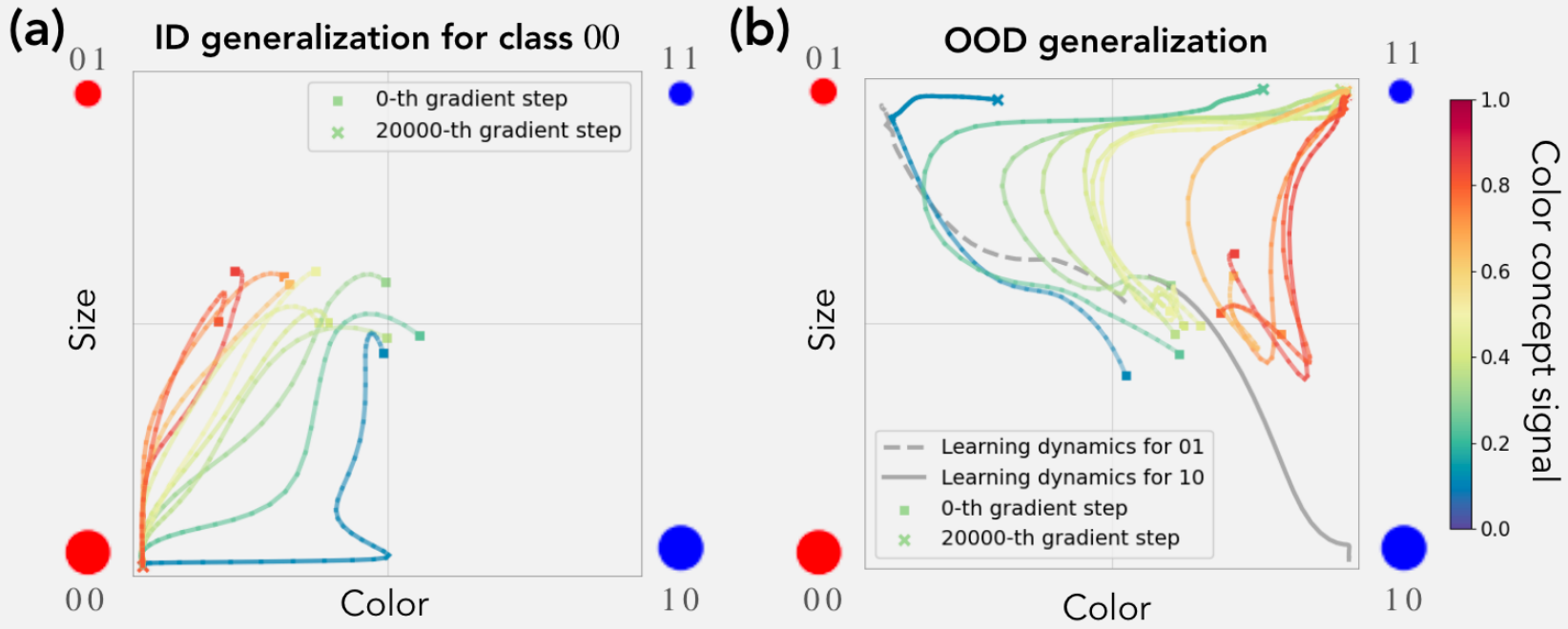
Generalization Dynamics

- An early stopped text-to-image model can witness concept memorization
  - simply associate an unseen conditioning to the nearest concept class when asked to generate OOD samples.

- Given sufficient time, the model will disentangle concepts underlying the data-generating process and learn to generate entirely novel, OOD samples.

Park et al., Emergence of Hidden Capabilities: Exploring Learning Dynamics in Concept Space, NeurIPS 2024

Generlization Dynamics



Park et al., Emergence of Hidden Capabilities: Exploring Learning Dynamics in Concept Space, NeurIPS 2024