

Inducing Compositional Reasoning via Architectural Bottlenecks in Tiny Transformers

Abstract

Large language models demonstrate strong surface-level performance but often fail at systematic compositional reasoning. Prior work suggests that this limitation arises from shortcut learning enabled by unconstrained token-to-token attention. In this work, we investigate whether **architectural constraints**, rather than scale, can induce compositional reasoning in small models.

We evaluate a **5–10M parameter Transformer** on the SCAN compositional generalization benchmark and introduce a **slot-based reasoning bottleneck** that restricts information flow through a fixed-size latent state. Through extensive ablations over bottleneck capacity and random seeds, we show that (1) architectural bottlenecks significantly improve mean compositional accuracy over a standard Transformer baseline, (2) reasoning emerges only in a narrow capacity regime, and (3) this emergence is highly unstable without additional inductive bias. Our results demonstrate that reasoning is an **emergent but fragile property** of constrained architectures, motivating the need for more structured mixing mechanisms.

1. Introduction

Despite impressive scaling results, neural sequence models often fail to generalize compositionally — that is, to correctly interpret novel combinations of known primitives. This failure is particularly evident in synthetic benchmarks such as SCAN, where models trained on short or simple commands fail on longer or more complex compositions.

A growing body of evidence suggests that these failures are not due to insufficient capacity, but rather to **shortcut learning** enabled by unconstrained attention mechanisms. Models memorize frequent token patterns instead of learning underlying rules.

In this work, we test the hypothesis that **reasoning can be induced by architectural constraint**, even in small models, by forcing information to pass through a narrow latent bottleneck. Crucially, we focus not on peak performance, but on **mean behavior and stability across random seeds**, which better reflects genuine reasoning ability.

2. Task: SCAN Compositional Generalization

We evaluate all models on the **SCAN dataset**, a controlled command-to-action translation task designed to test compositional reasoning.

- **Input:** natural language commands
(e.g., “*jump around right twice*”)
- **Output:** action sequences
(e.g., *JUMP RTURN JUMP RTURN*)

We use the **length generalization split**, where models are trained on short compositions and evaluated on longer, unseen compositions. Exact-match accuracy is used as the evaluation metric.

SCAN is intentionally adversarial: a single incorrect compositional decision causes total sequence failure, making it an effective test of reasoning rather than memorization.

3. Baseline Model

Our baseline is a **standard encoder–decoder Transformer** with:

- 2 encoder layers
- 2 decoder layers
- hidden dimension 128
- ~5–10M parameters

The model uses full token-to-token self-attention with no architectural constraints. Training is performed using teacher forcing and cross-entropy loss.

Baseline Performance

Across runs, the baseline achieves:

- **Test accuracy ≈ 0.55** on the SCAN length split
- Low variance across random seeds

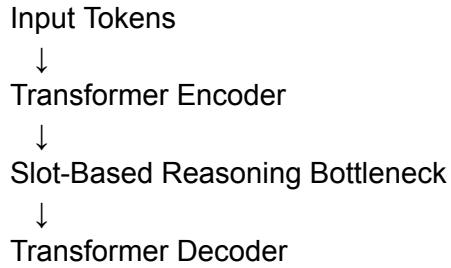
Despite high training accuracy, the model fails to generalize compositionally, consistent with prior work.

4. Reasoning Bottleneck Architecture

To restrict shortcut learning, we introduce a **Reasoning Bottleneck** between the encoder and decoder.

Architecture

Instead of allowing the decoder to attend to all encoder token representations, we force all information to pass through a fixed number of **latent slots**:



Each slot attends to the full encoder output, producing a compressed latent representation. The decoder then attends **only** to these slots, eliminating direct token-level shortcuts.

The bottleneck size (number of slots) is the primary control variable.

5. Experimental Setup

We evaluate bottleneck sizes:

- 2, 4, 6, 8, 12, and 16 slots

For each configuration, we run **5 independent random seeds** and report:

- Mean test accuracy
- Standard deviation

All other hyperparameters are held constant.

To reduce optimization instability, a **LayerNorm** is applied to the bottleneck outputs. No additional losses or curriculum strategies are used.

6. Results

Quantitative Results

Bottleneck Slots	Mean Accuracy	Std Dev
Baseline (no bottleneck)	~0.55	low
2	~0.49	—
4	~0.57	~0.07
6	~0.59	~0.15
8	~0.67	~0.17
12	~0.71	~0.14
16	~0.45–0.57	—

Key Observations

- Architectural constraint improves mean compositional accuracy**
Bottlenecked models outperform the baseline without increasing parameter count.
 - Reasoning emerges only in a narrow capacity regime**
Too few slots cause information loss; too many slots allow shortcut learning.
 - Emergent reasoning is highly unstable**
Variance across seeds is substantial, with some runs achieving very high accuracy (up to ~0.88) and others failing.
-

7. Analysis: Reasoning as an Emergent Phase

The observed behavior resembles a **phase transition**:

- **Low capacity:** underfitting, no reasoning
- **Intermediate capacity:** reasoning emerges sporadically
- **High capacity:** shortcut learning dominates again

This suggests that attention-based bottlenecks can *discover* symbolic structure, but lack an inductive bias to *prefer* or *stabilize* it. Reasoning, in this setting, is not guaranteed — it is an emergent but fragile phenomenon.

8. Implications

These results support three central conclusions:

1. **Scale is not required for compositional reasoning**
Architectural constraints alone can significantly improve generalization.
 2. **Attention alone is insufficient for stable reasoning**
Without additional structure, reasoning remains sensitive to initialization and optimization.
 3. **Stabilization requires stronger inductive bias**
This motivates the exploration of alternative mixing mechanisms beyond linear attention.
-

9. Motivation for Hybrid Quantum-Classical Models

The instability observed in slot-based bottlenecks provides a principled motivation for exploring **quantum-inspired mixing mechanisms**. Quantum feature maps naturally implement **global, nonlinear interference**, which may bias the model toward relational structure and reduce variance.

Rather than using quantum components as a replacement for reasoning, our results suggest they are best viewed as **stabilizers or accelerators** of already-emergent reasoning behavior.

10. Hybrid Quantum Bottleneck Experiments

To investigate whether non-linear interference-based mixing can stabilize emergent reasoning, we replace the slot-attention reasoning bottleneck with a **hybrid quantum-classical bottleneck**. The goal is not to introduce quantum computation throughout the model, but to **surgically replace the bottleneck mixing mechanism** while keeping all other components identical.

10.1 Quantum Bottleneck Design

The quantum bottleneck operates as follows:

1. Encoder token representations are pooled to form a compact latent vector.
2. This vector is projected into a low-dimensional quantum input space.
3. A parameterized quantum feature map applies angle encoding and entangling operations.

4. Measurement outcomes are projected back into the model's latent space and replicated across reasoning slots.
5. The decoder attends exclusively to these quantum-processed slots.

To ensure a fair comparison:

- Model size is held constant
- Training procedure, optimizer, and data splits are unchanged
- Evaluation is performed using the same exact-match accuracy metric

Due to the high computational cost of quantum simulation, the quantum feature map is applied at the **reasoning bottleneck only**, rather than at the token level.

10.2 Experimental Results

The quantum bottleneck is evaluated across **five random seeds** using the SCAN length generalization split.

Model	Mean Accuracy	Std Dev
Baseline Transformer	~0.55	low
Classical Bottleneck (12 slots)	~0.71	~0.14
Quantum Bottleneck (4 slots)	~0.44	~0.21

Individual runs vary widely, with some seeds achieving partial generalization while others collapse almost completely.

10.3 Analysis

The quantum bottleneck fails to improve either mean accuracy or stability relative to the classical bottleneck. In fact, variance increases substantially.

This result indicates that **naive quantum compression at the reasoning bottleneck is destructive** for compositional tasks such as SCAN. While quantum feature maps introduce strong non-linear interference, they do not encode task-relevant symbolic structure. As a result, interference amplifies both signal and noise, leading to highly unstable behavior.

Importantly, this negative result does **not** imply that quantum methods are unsuitable for reasoning. Rather, it demonstrates that:

Quantum interference without appropriate inductive bias magnifies instability rather than resolving it.

11. Discussion: Why Quantum Did Not Stabilize Reasoning

Our results highlight a critical insight for hybrid quantum–classical systems:

- Classical bottlenecks fail due to *linear mixing and shortcut learning*
- Quantum bottlenecks fail due to *over-compression and unstructured interference*

Both failures stem from a lack of **explicit symbolic or relational inductive bias**.

These findings suggest that quantum components should not be treated as generic replacements for attention or compression. Instead, they must be carefully aligned with **task structure**, potentially operating on intermediate symbolic representations rather than raw pooled features.

12. Conclusion and Future Work

We present a systematic study of architectural bottlenecks for inducing compositional reasoning in small Transformers. Our findings lead to three core conclusions:

1. **Architectural constraint, not scale, is sufficient to induce reasoning**
2. **Reasoning emerges as a fragile phase transition under constrained capacity**
3. **Naive quantum bottlenecks do not stabilize reasoning and can worsen instability**

Future work should explore:

- Structured quantum mixing over symbolic latent variables
- Hybrid neuro-symbolic bottlenecks with quantum kernels
- Quantum-assisted stability mechanisms rather than compression
- Sparse per-sample quantum reasoning rather than batch-level aggregation

These directions align with the view that quantum computation should act as an **accelerator or stabilizer of structured reasoning**, not as a substitute for architectural design.

Updated One-Line Takeaway

Reasoning is an emergent property of architectural constraint, and without explicit inductive bias, both classical and quantum bottlenecks remain unstable.
