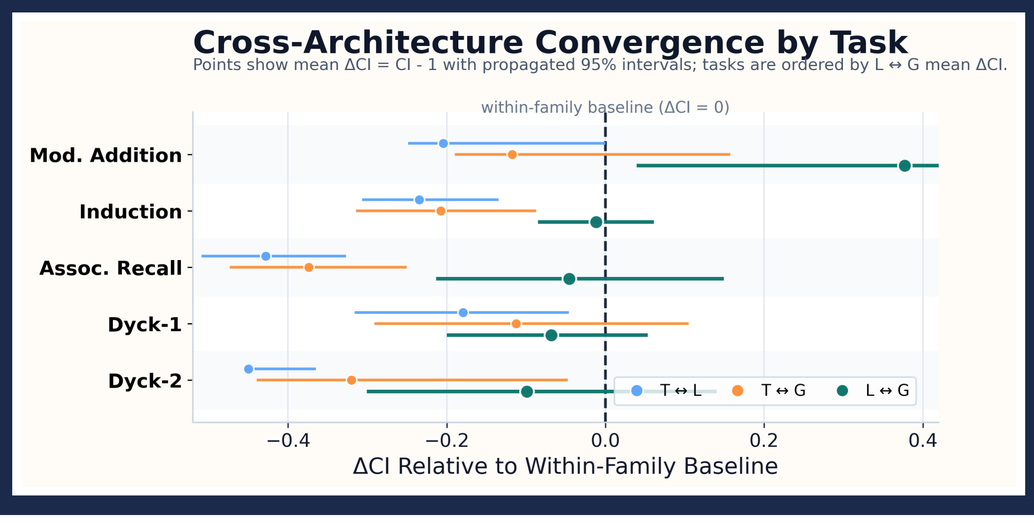


Convergence Is Selective, Not Universal

Recurrent models (LSTM, GRU) learn aligned internal representations on every task we tested. Transformers diverge.

ICML International Conference On Machine Learning

Cross-Architecture CKA Reveals Where Sequence Models Converge and Diverge on Synthetic Mechanistic Tasks

Harsh Rathva ▪ Sardar Vallabhbhai National Institute of Technology, India ▪ u24ai036@aid.svnit.ac.in

1 TL;DR

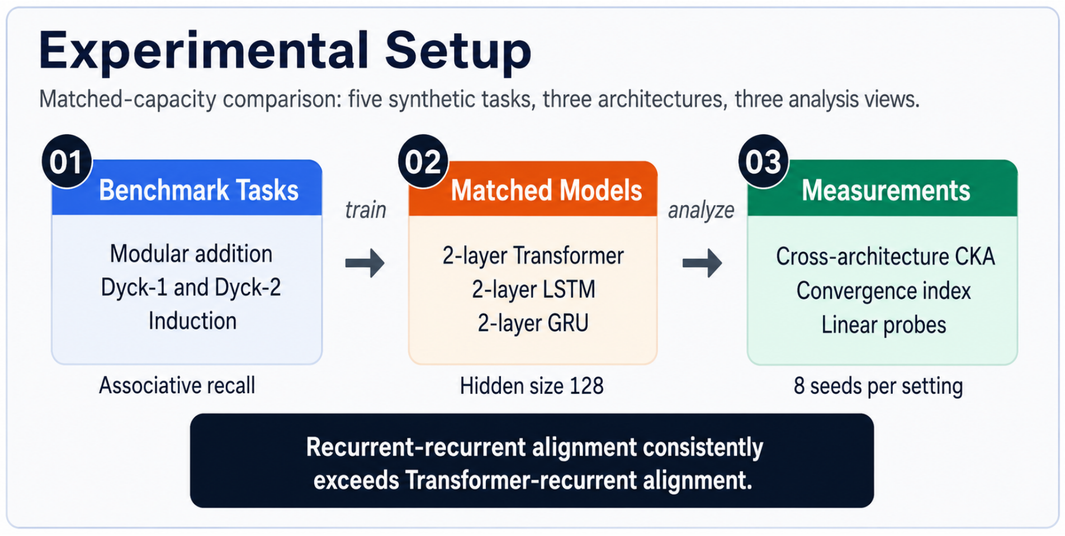
MAIN TAKEAWAYS

- **Asymmetric competence creates interpretive hazards.** Under identical training, recurrent models (LSTM, GRU) substantially outperform Transformers on all five tasks. CKA between a competent and an incompetent model reflects shared input geometry, **not** algorithmic convergence.
- **Within-family alignment dominates.** LSTM↔GRU convergence indices reach 0.90–1.38; Transformer↔recurrent indices drop as low as **0.55** on hierarchical tasks (Dyck-2).
- **Geometric similarity ≠ shared algorithms.** Moderate CKA persists at embedding layers even at chance-level performance, driven by tokenization structure rather than learned computation.

2 Background & Setup

Motivating question: When matched-capacity sequence models train on the same task, do they converge to aligned internal representations?

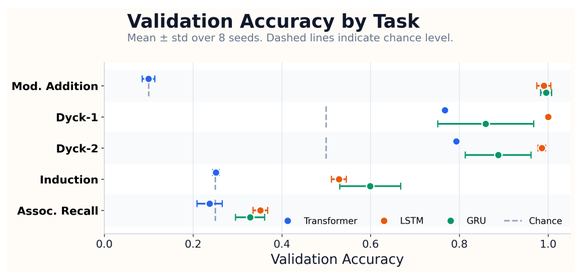
We train **two-layer Transformer, LSTM, and GRU** classifiers (hidden dim 128) on five synthetic tasks with **identical hyperparameters** and **8 random seeds** per setting.

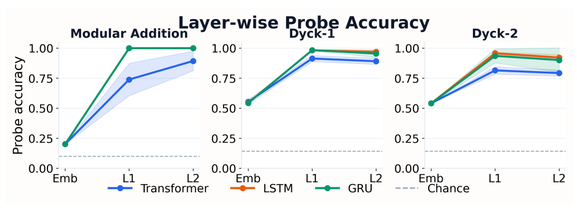


Analysis pipeline:

- Layer-wise **linear CKA** on mean-pooled activations
- **Convergence index:** cross-arch CKA normalized by within-arch seed baselines
- Layer-wise **linear probes** for task-relevant features

3 Task Performance



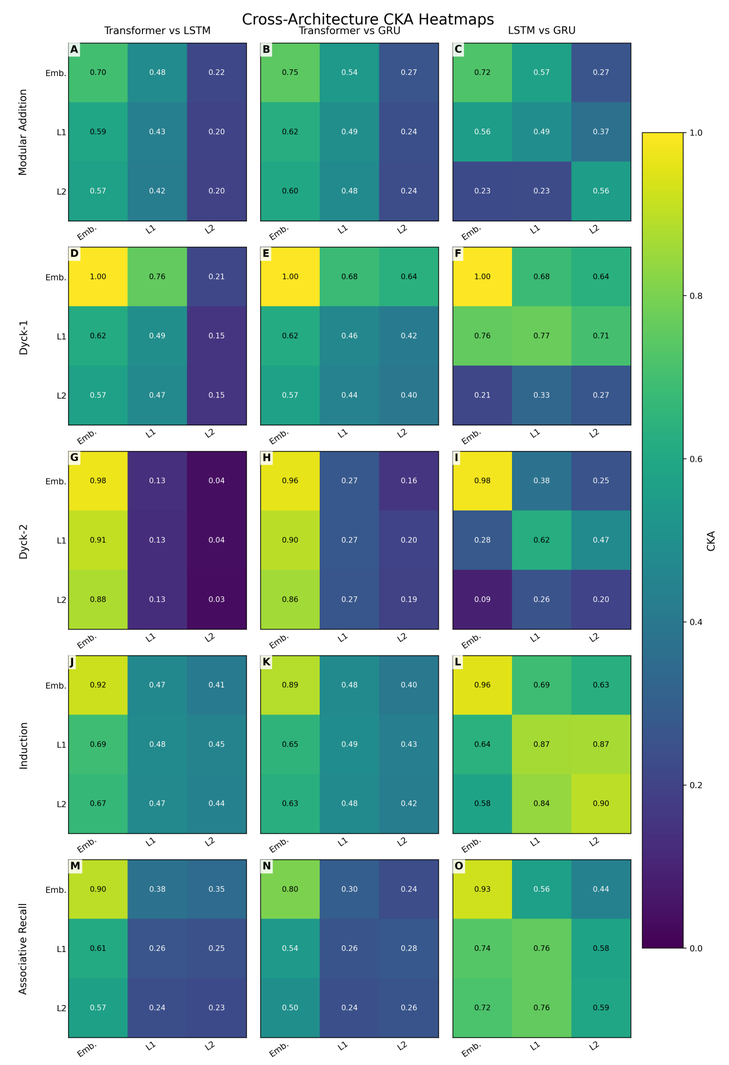


Key observations:

- Recurrent models solve modular addition and Dyck-1 near-perfectly (>99%)
- Transformer is **chance-level** on modular addition (0.10 for 10 classes)
- Asymmetric competence is **intentional**: it shows how CKA can mislead when models differ in competence

Linear probes show the chance-level Transformer still encodes partial task structure, separating **representational content** from **readout competence**.

4 CKA & Convergence



Three patterns emerge:

- **Embedding CKA is universally high** (0.70–0.98), reflecting shared tokenization geometry, not learned computation
- **Deeper-layer CKA decays fastest** for Transformer–recurrent pairs, especially Dyck-2 (T↔LSTM L2: **0.03**)
- **LSTM–GRU alignment stays strong** at all layers (CI ≥ 0.90), consistent with shared recurrent inductive biases

Convergence index:

- L↔G CI ≈ 0.90–1.38: cross-family alignment **matches or exceeds** within-family seed consistency
- T↔L CI drops to **0.55 on Dyck-2**: architecture-specific geometry for hierarchical tasks

5 Limitations & Outlook

- **Transformer training:** Pilot grokking (500 ep.) raises acc. to 0.94, T↔L CKA from 0.20 → 0.68
- **Scale:** Two-layer models (dim 128), synthetic tasks; generalization to larger models open
- **Methodology:** Mean-pooled activations may miss token-level structure; CKA confounded by dataset geometry
- **Next steps:** Full-seed grokking, activation patching with controls, state-space models (Mamba)

Implication for mechanistic interpretability:
Mechanistic explanations transfer more reliably **within** architectural families (LSTM → GRU) than **across** (Transformer → LSTM). CKA is best used as a **triage tool**, not standalone evidence of mechanistic alignment.