A Neuroscience-Inspired Dual-Process Model of Compositional Generalization

Alex Noviello[1*], Claas Beger[1*], Jacob Groner[1], Kevin Ellis[1†], Weinan Sun[2†]

[1]Cornell University, Department of Computer Science, Ithaca NY, USA
[2]Cornell University, Department of Neurobiology and Behavior, Ithaca NY, USA

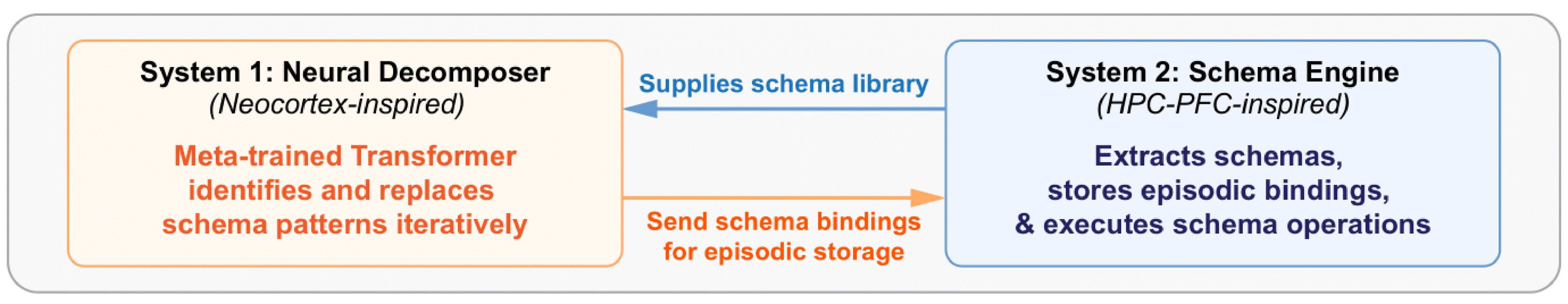
* Equal contribution.
† Equal advising

BACKGROUND

Large Language Model struggle with tasks requiring composition

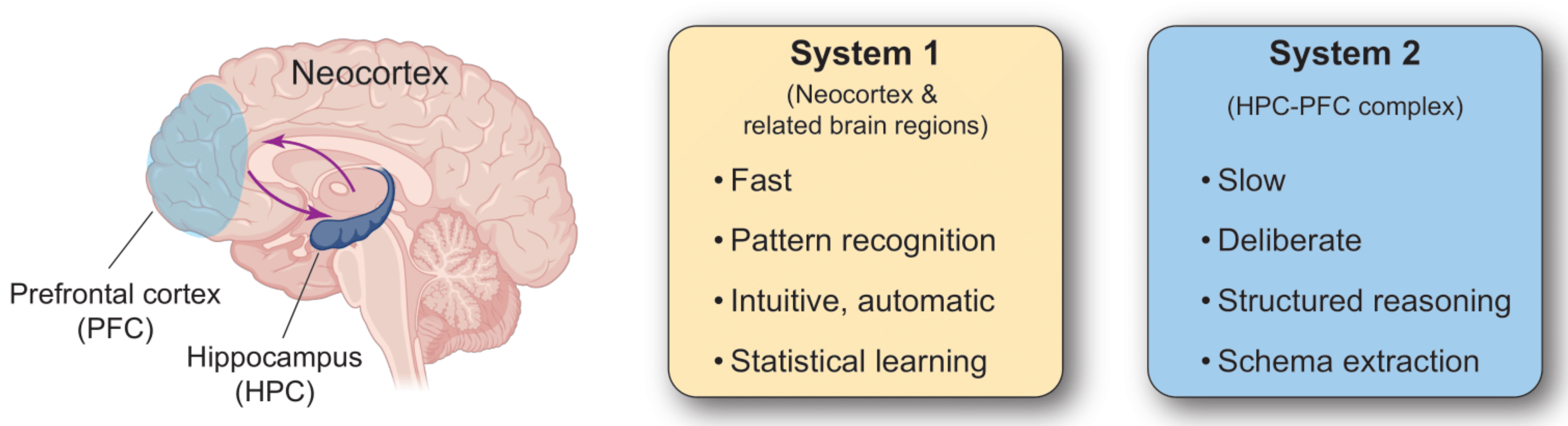
- Even State-of-the-Art models tend to **learn shallow heuristics instead of abstract rules**, which prevents them from performing proper compositional generalization
- Yet, humans can do so effortlessly, generalizing by composing known concepts in new ways, a capacity termed **systematic compositionality**
- From a neuroscientific standpoint, this is just one of the abilities likely deriving from **the interplay of complementary learning systems (CLS)**

MIRAGE ARCHITECTURE



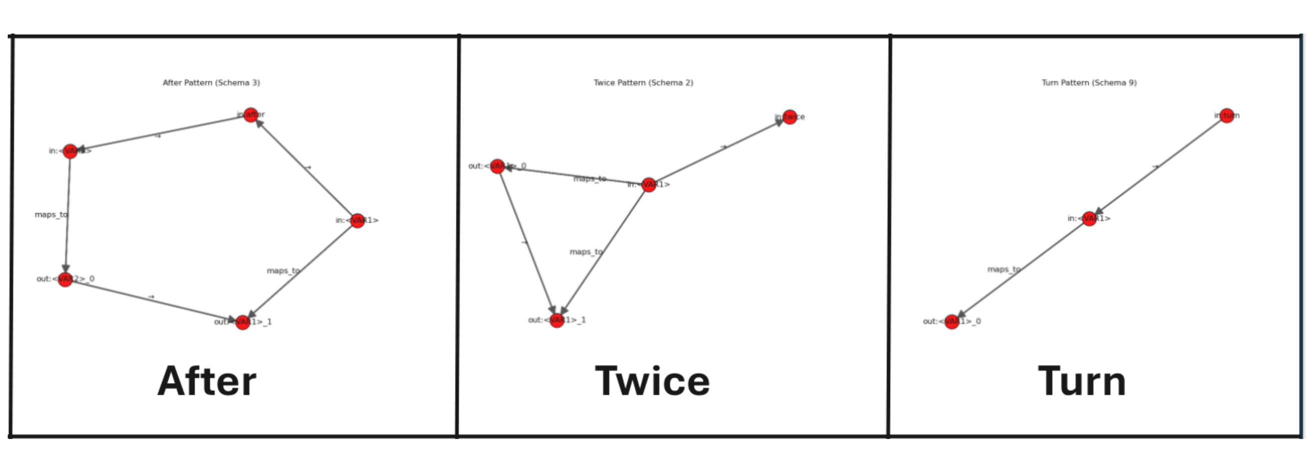
We realize our System MIRAGE (Meta-Inference using Rules and Abstractions from Generalized Experience) through a small Transformer model, which marks and resolves schemas iteratively and an explicit schema learner that fits distinct rules to given sequences (experiences)

NEUROSCIENCE INSPIRATION



- **The brain balances rapid learning and slow abstraction**: the hippocampus quickly encodes episodic experiences while the neocortex extracts long-term regularities, supporting both fast adaptation and stable, intuitive (System 1) processing.
- **Deliberate reasoning (System 2) arises from hippocampus–PFC interactions**, where episodic memories are bound into cognitive maps and progressively abstracted into reusable schemas that enable flexible problem-solving and knowledge recombination.

SCHEMA REPRESENTATIONS



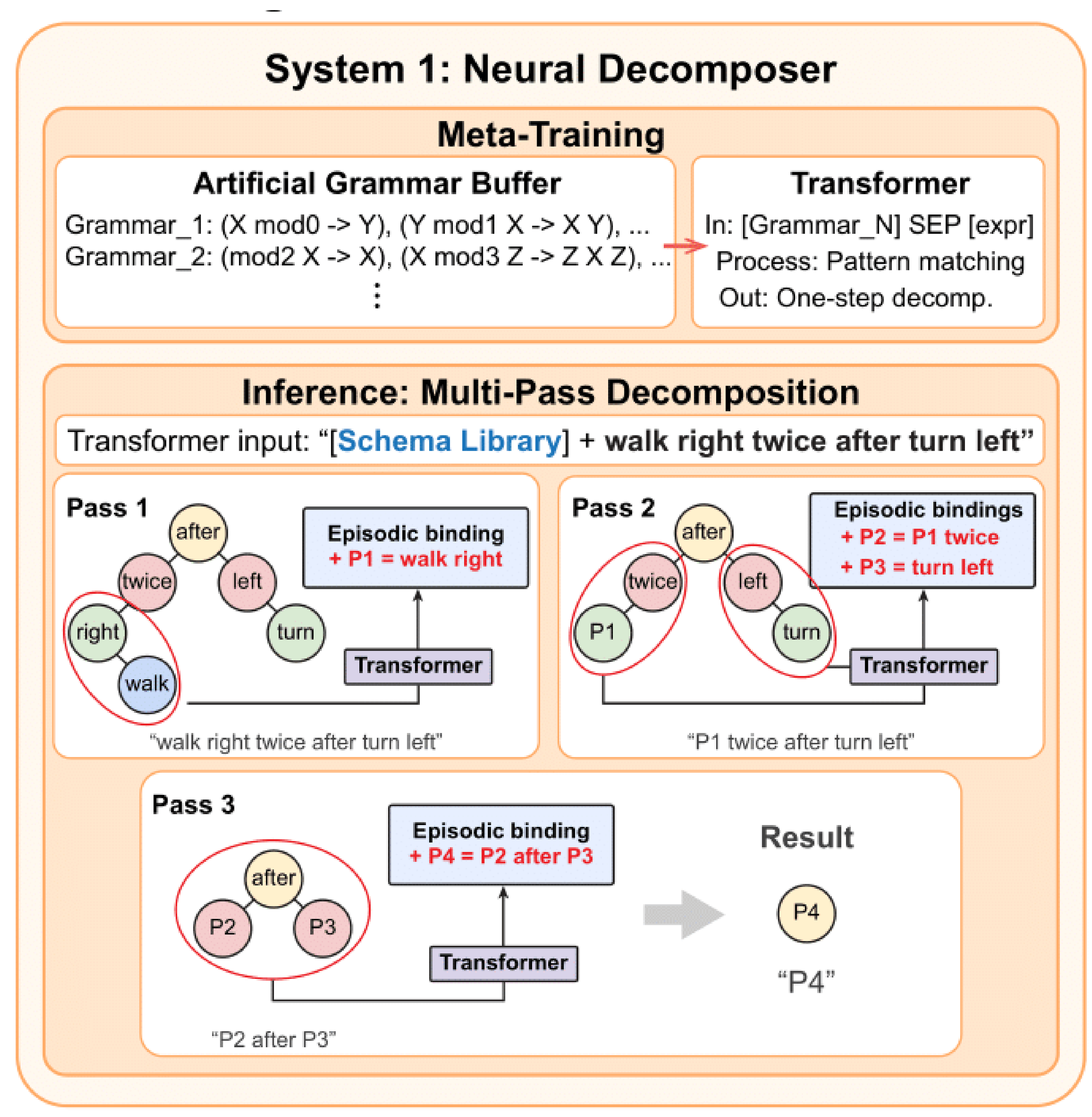
Schemas can be represented in various ways. We primarily present results using Causal Structured Clone Graphs (CSCGs) which can express simple parameterized functions and was explicitly designed as a model of cognitive maps.

Claas Beger

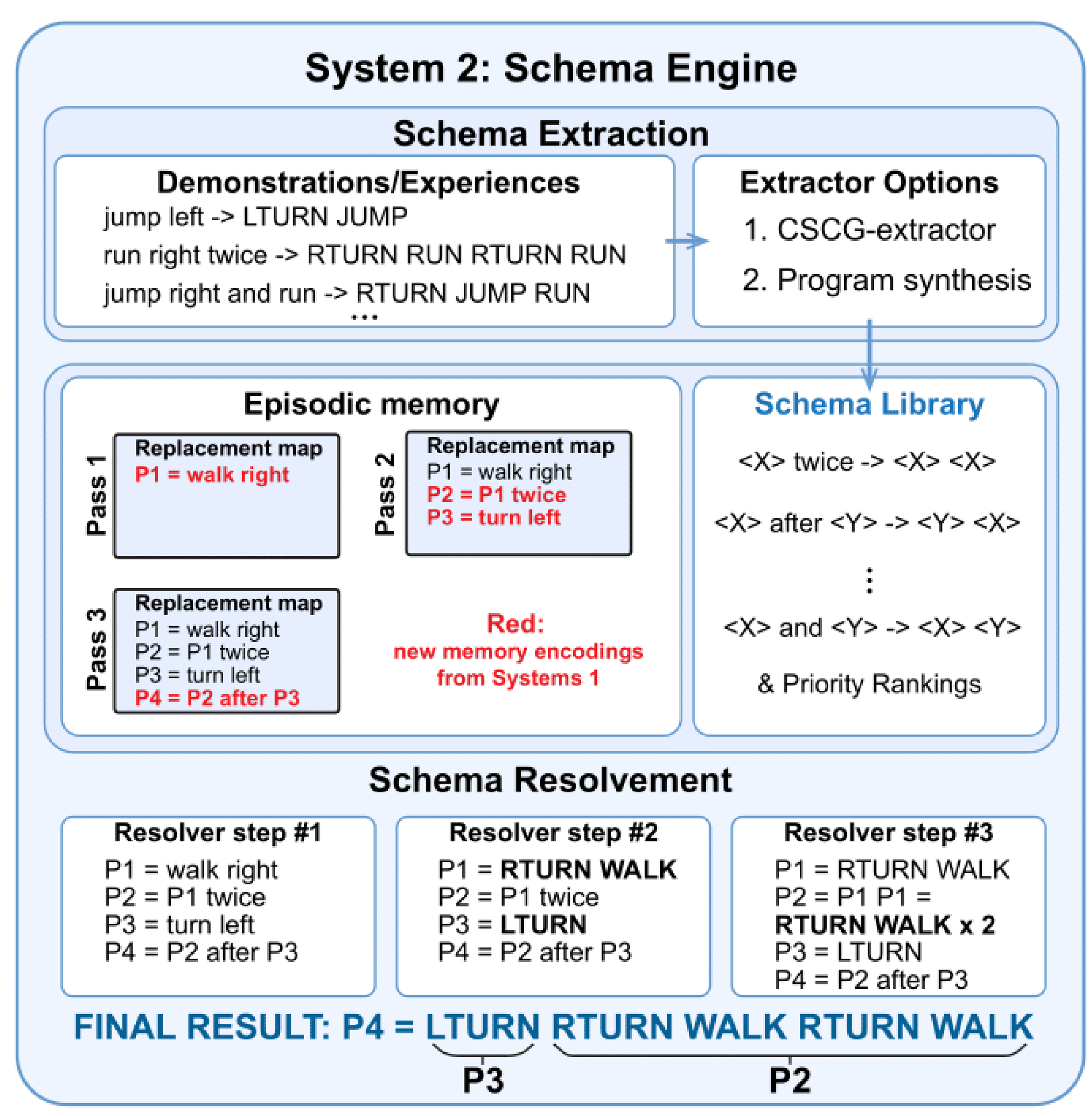
NeuroAI Lab

Paper

IMPLEMENTATION & TRAINING/INFERENCE PROCEDURE



- **A standardized, anonymous vocabulary lets any grammar be converted into a shared token format**, enabling the transformer to interpret primitives, modifiers, and schemas in a uniform way.
- **By continually generating and sampling random grammars, the model learns to parse and recombine fresh symbolic structures**, producing sequences that explicitly mark schema applications and preventing memorization of any single grammar.



- The **engine builds a grammar from examples** by systematically proposing simple template edits (like span replacements or re-orderings) and keeping only those templates that improve corpus-wide consistency and accuracy; accepted templates are translated to CSCGs, with precedence inferred from how often one schema applies before another.
- **This iterative process continues until no better rules can be found**, producing a grammar (P, M, Σ, π) ;the proposal mechanism is modular and also compatible with alternative approaches (for example Program Synthesis)

RESULTS

		SCAN splits			
Model	**Full Task**	**Simple**	**Length**	**Add prim. 'jump'**	**Template**
Transformer	N/A	99.85 ± 0.00	13.58 ± 0.01	0.40 ± 0.13	3.09 ± 3.01
Transformer+SC_Library	N/A	99.91 ± 0.11	15.86 ± 1.36	0.03 ± 0.04	0.00 ± 0.00
MIRAGE (ours)	99.59 ± 0.24	99.50 ± 0.30	99.35 ± 0.41	99.65 ± 0.20	99.55 ± 0.23

- We evaluate our System on the SCAN benchmark, which creates **compositional statement from simple navigation orders**
- SCAN also provides splits excluding certain attributes (single commands, length
- We further compare against a **vanilla Transformer**, with or without given the **Schema Library in-Context**
- MIRAGE solves all splits with **near perfect performance**