Extended Abstract Track

Transformers Represent Causal Abstractions

Editors: List of editors' names

Abstract

Agents often interact with environments too complex to model in microscopic detail. Abstractions offer a way to form useful models anyway. When and how do such abstractions arise? Drawing on recent work on macro-level structure ("emergence") in complex systems, we hypothesize that agents interacting with such systems naturally learn abstractions aligned with the macro-level. To investigate, we introduce a parameterized hidden Markov model (HMM) with a tunable degree of macro-structure. We then train a transformer on sequences of observables generated by the HMM and track the evolution of abstractions represented in its residual stream. As the macro-structure parameter is varied, we observe systematic changes in internal representations and dynamics. These results provide preliminary evidence that exposure to macro-structured processes drives the emergence of abstractions in deep models.

Keywords: Abstraction; emergence; coarse-graining; lumpability; Hidden Markov models; transformers; representation learning.

1. Introduction

In this work we investigate how a neural network internally develops its model of the world. We begin with the notion that an intelligent agent must develop a world model to understand what initially appears to be a disordered world. The literature suggests that a sufficiently adaptive agent "must have learned a causal model of the data-generating process" Jonathan Richens (2024). However, we also expect that agents do not always learn the exact microscopic structure of hidden dynamics, but instead learn useful abstractions Allen et al. (2024).

We work within the framework of next-observable prediction, with a neural network trained on the outputs determined from hidden dynamics, and we call these useful abstractions that neural networks use to learn such next-token prediction "causal abstractions". We are interested in the formation and evolution of these causal abstractions.

In this setting, we analyze the learning process when the hidden markov model displays conditions of lumpability and quasi-lumpability. Lumpability is satisfied when there exist coarse-grainings of states such that its distributions do not break the markov property. Quasi-lumpability is satisfied when there exist coarse-grainings of states such that its distributions hold for some tolerance delta. We care about lumpability conditions because during early training, it remains ambiguous as to whether or not a coarse-graining will be the best to use to model the hidden dynamics, and therefore, predict the next observable. In our setting, the coarse-grained model corresponds to a macroscaled causal abstraction, and a non-coarse-grained model corresponds to the microscale.

Using techniques found in Shai et al. (2024), we are able to probe transformers for linear representations of belief states over hidden state structure. We analyze the development of these representations, and we present preliminary evidence for abstraction formation, and find qualitative changes in learning dynamics in the lumpable and quasi-lumpable settings.

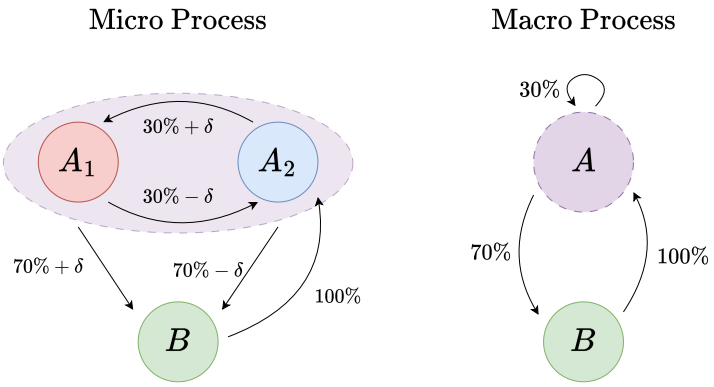

Figure 1: We illustrate `Lumpy3`, a parameterized hidden Markov model with three hidden states, shown left. The parameter δ controls "lumpability". Exactly when $\delta = 0$, hidden states A_1 and A_2 may be "coarse-grained" into a single state A, forming a *causally-closed* macro process, shown right.

2. Background

A *Hidden Markov Model* (HMM) is a pair of stochastic processes (Z_t, X_t), where Z_t is called the hidden process and X_t is called the observable process. The transition dynamics between hidden states satisfy the Markov property $P(Z_{t+1} \mid Z_1, \ldots, Z_t) = P(Z_{t+1} \mid Z_t)$, and the emission dynamics of the observations satisfy conditional independence given the hidden states $P(X_t \mid Z_1, \ldots, Z_t, X_1, \ldots, X_{t-1}) = P(X_t \mid Z_t)$. The term "hidden" corresponds to the fact that the ground-truth states are not directly accessible to the observer; the observer is instead, however, presented with the sequence of observables, $\{x_{0:t}\}$.

HMMs are a natural testbed for studying world models in neural networks. Shai et al. (2024) trained transformers for next-observable prediction, $P(X_{t+1} \mid X_{1:t})$, on sequences generated by an HMM. Rather than capturing only surface statistics, the models learned linear representations—visible in the residual stream—of beliefs over the unobserved hidden states. These beliefs matched the ground-truth posteriors, $P(Z_t \mid X_{1:t})$, computed via Bayes' rule from the HMM's transition and emission dynamics—information never provided to the model.

Lumpability refers to a Markov process remaining Markovian after coarse-graining. For a Markov chain $\{Z_t\}$ on a finite state space S with transition matrix P. S is partitioned into k disjoint subsets $S = C_1 \cup C_2 \cup \ldots \cup C_k$. The coarse-grained process $\{\tilde{Z}_t\}$, where $\tilde{Z}_t = i$ whenever $Z_t \in C_i$, is itself a Markov chain. (Strong) lumpability refers to those processes which remain Markovian regardless of initial conditions; quasi-lumpability refers to Markov processes that are not lumpable, but are nearly so (Franceschinis and Muntz, 1994); see Appendix A.

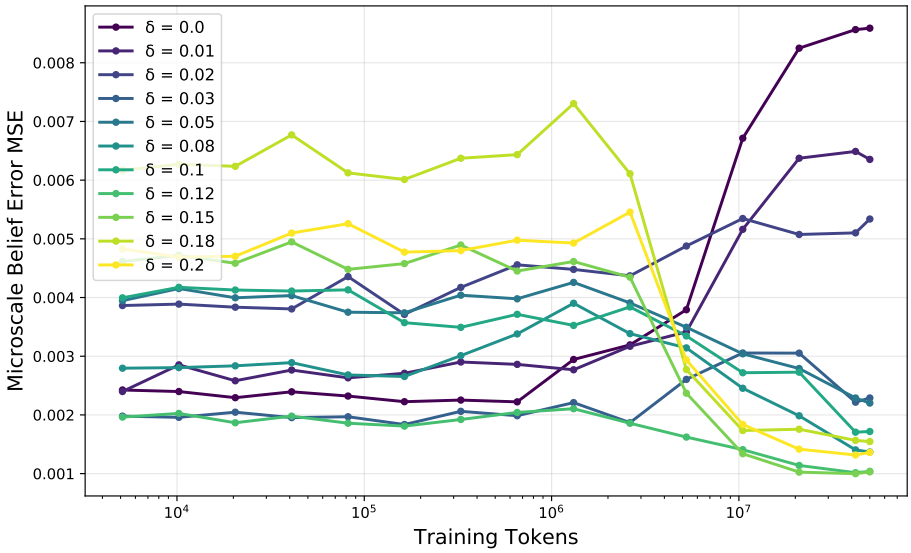

Figure 2: We train a transformer on "next-observable prediction" over independent training runs while varying a lumpability parameter δ. We expect the transformer to learn macrolevel structure for $\delta \approx 0$ and microlevel structure for $\delta \gg 0$. We observe qualitatively different behavior between these low and high δ regimes, possibly corresponding to learning the respective levels.

3. Methods

We train a transformer on sequences of observables emitted by a parameterized 3-state HMM `Lumpy3`, depicted in Figure 1. Paths through the process emit sequences of observables: binary symbols a, and b, following the deterministic conditions: $P(X = a \mid Z = A_1) = P(X = a \mid Z = A_2) = 1$ and $P(X = b \mid Z = B) = 1$. We distinguish between the hidden states A_1 and A_2 by establishing that B is invariably followed by A_2.

Furthermore, for $\delta = 0$, `Lumpy3` is *strongly lumpable*, as the coarse-graining map maintains the Markov property, and is causally closed. As shown in the appendix, $\delta = 0$ corresponds to causal closure as the microstates A_1 and A_2 have equivalent dynamics in relationship to one another, and to state B_2; this parameter is dynamically reducible to its macroscale without loss of relevant information. A non-zero δ breaks the symmetry condition for strong lumpability, because the coarse-grain map which collapses A_1 and A_2 into A is no longer dynamically consistent with the micro-scale.

For each value of p, δ, and each training checkpoint, we analyzed the model's ability to recover ground-truth belief states. First we generated 100 sequences generated from paths in `Lumpy3`. We then extracted residual stream activations from all transformer layers. We then used linear regression to predict ground-truth belief states from activations, and computed the mean-squared error between predicted and actual ground-truth belief states of the ϵ-machines belonging to each `Lumpy3` process.

For high values of δ, we expect transformers trained on `Lumpy3` to rely on macroscale abstraction early in the training process. However, larger δ values break lumpability and therefore reduce the mutual predictive information at the macroscale trajectories. In this setting the macroscale is not causally closed: predictive information about the future is lost under the coarse-graining map.

4. Results

For higher values of $\delta \gg 0$ such as $\delta = 0.20$, 0.18, and 0.15, the microscale MSE remains stable, decreases, then remains stable again. For intermediate values of δ, we observe that the microscale MSE increases over learning, and then decreases. For the lower values of $\delta \approx 0$ such as 0.01 or 0.02, as well as $\delta = 0$, the microscale MSE also remains relatively constant, increases sharply, and remains stably constant again.

5. Discussion

We hypothesize that the regions where the function remains flat, as well as the regions of pronounced increases or decreases in the MSE between ground-state truth and predictive belief states (especially for lower and higher values of δ) is evidence in support of the existence of phase changes of causal abstraction, analogous to dynamical phase changes. The regions of MSE constancy, we suggest, may be regions of equilibria; meanwhile, the pronounced increases and decreases in MSE suggest regions in which the transformer begins to prefer one abstraction. These opposite shifts may be the result of competing pressures that govern representational preference. There is a pressure associated with the compression of trajectories into a causally closed coarse-grained description, which dominates for low values of δ where the macroscale nearly preserves predictive information. On the other hand, there is a pressure to accurately track the microscale dynamics, which dominates at high values of δ where the macroscale leaks predictive information. The phase changes we observe correspond to critical points where the balance between these two forces tips, causing the model to reorganize its internal geometry. Discard too much information, and the resulting representation cannot accurately characterize the environment. In this sense, the phase diagram of training reflects a conservation-like principle: the transformer cannot simultaneously maximize both simplicity and fidelity, and instead trades off between them as a function of δ and training time.

References

Cameron Allen, Neev Parikh, Omer Gottesman, and George Konidaris. Learning markov state abstractions for deep reinforcement learning, 2024. URL https://arxiv.org/abs/2106.04379.

Giuliana Franceschinis and Richard R Muntz. Bounds for quasi-lumpable markov chains. *Performance Evaluation*, 20(1-3):223–243, 1994.

Tom Everitt Jonathan Richens. Robust agents learn causal world models, 2024. URL https://arxiv.org/abs/2402.10877.

Fernando E. Rosas, Bernhard C. Geiger, Andrea I Luppi, Anil K. Seth, Daniel Polani, Michael Gastpar, and Pedro A. M. Mediano. Software in the natural world: A computational approach to hierarchical emergence, 2024. URL https://arxiv.org/abs/2402.09090.

Adam Shai, Lucas Teixeira, Alexander Oldenziel, Sarah Marzen, and Paul Riechers. Transformers represent belief state geometry in their residual stream. *Advances in Neural Information Processing Systems*, 37:75012–75034, 2024.

Appendix A. Quasi-lumpability

Formally, we say that if transitions within the same block C_i lead to similar distributions over the coarse-grained states, then the chain is *quasi-lumpable* with tolerance $\delta \geq 0$ if for all C_i, C_j:

$$\max_{x,x' \in C_i} \left| \sum_{s \in C_j} P(x,s) - \sum_{s \in C_j} P(x',s) \right| \leq \delta.$$

Appendix B. Computational Mechanics

From computational mechanics, we borrow the vocabulary of ϵ-machines. We group together past-trajectories into causal states if they have equivalent future statistics. The causal states of a time series $X = \{X_t\}_{t \in \mathbb{N}}$ are equivalence classes of past trajectories $\boldsymbol{x}_t := (\ldots, x_{t-1}, x_t)$ established by $\boldsymbol{x}_t \equiv_\epsilon \boldsymbol{x}'_t$ iff $p\big(\boldsymbol{x}^L_{t+1} \mid \boldsymbol{x}_t\big) = p\big(\boldsymbol{x}^L_{t+1} \mid \boldsymbol{x}'_t\big)\,, \quad \forall\, \boldsymbol{x}^L_{t+1},\, L \in \mathbb{N}$. We coarse-grain past trajectories with a mapping, ϵ, thus generating a new time series of causal states $E = \{E_t\}_{t \in \mathbb{N}}$ with $E_t = \epsilon(\boldsymbol{X}_t)$.

Appendix C. Analysis of `Lumpy3`

We write the unique casual states for `Lumpy3`. There are only four. All sequences of observables fall into one of four classes, which we assign to a causal state e, representing an equivalence class. The causal states are given by

$$\begin{aligned}
e_0 &= \epsilon(X_{0:t} = a^k) && k \geq 1 \\
e_1 &= \epsilon(X_{0:t} = \cdots b) && \\
e_2 &= \epsilon(X_{0:t} = \cdots b a^k) && k \geq 1, even \\
e_3 &= \epsilon(X_{0:t} = \cdots b a^k) && k \geq 1, odd
\end{aligned}$$

That these are indeed the causal states can be verified by the fact that all sequences within an equivalence class share a corresponding distribution over observables, which we provide for each class.

$$\begin{aligned}
P(X_{t+1}|e_0) &= \begin{cases} 1-p & X_{t+1} = a \\ p & X_{t+1} = b \end{cases} \\
P(X = a|e_1) &= 1 \\
P(X_{t+1}|e_2) &= \begin{cases} 1-p+\delta & X_{t+1} = a \\ p-\delta & X_{t+1} = b \end{cases} \\
P(X_{t+1}|e_3) &= \begin{cases} 1-p-\delta & X_{t+1} = a \\ p+\delta & X_{t+1} = b \end{cases}
\end{aligned}$$

For $\delta = 0$, causal states $e_0 = e_2 = e_3$, forming an epsilon machine which is equivalent to the epsilon machine of macro-process displayed right in Figure 1. This epsilon machine equivalence is exactly the condition for causal closure defined by Rosas et al. (2024).

Appendix D. Hyperparameters

`Lumpy3` **parameters**	
p	0.5
δ	$\{0.0, 0.01, 0.02, 0.03, 0.05, 0.08, 0.1, 0.12, 0.15, 0.18, 0.2\}$
Training setup	
Training Tokens	50,000,000
Learning Rate	1×10^{-4}
Weight Decay	0.0
Model parameters	
d_{vocab}	2
d_{model}	64
n_{ctx}	10
d_{head}	8
n_{head}	1
d_{mlp}	256
n_{layers}	4

Table 1: All hyperparameters used for experiments.