How symmetry shapes abstraction via hierarchical beliefs

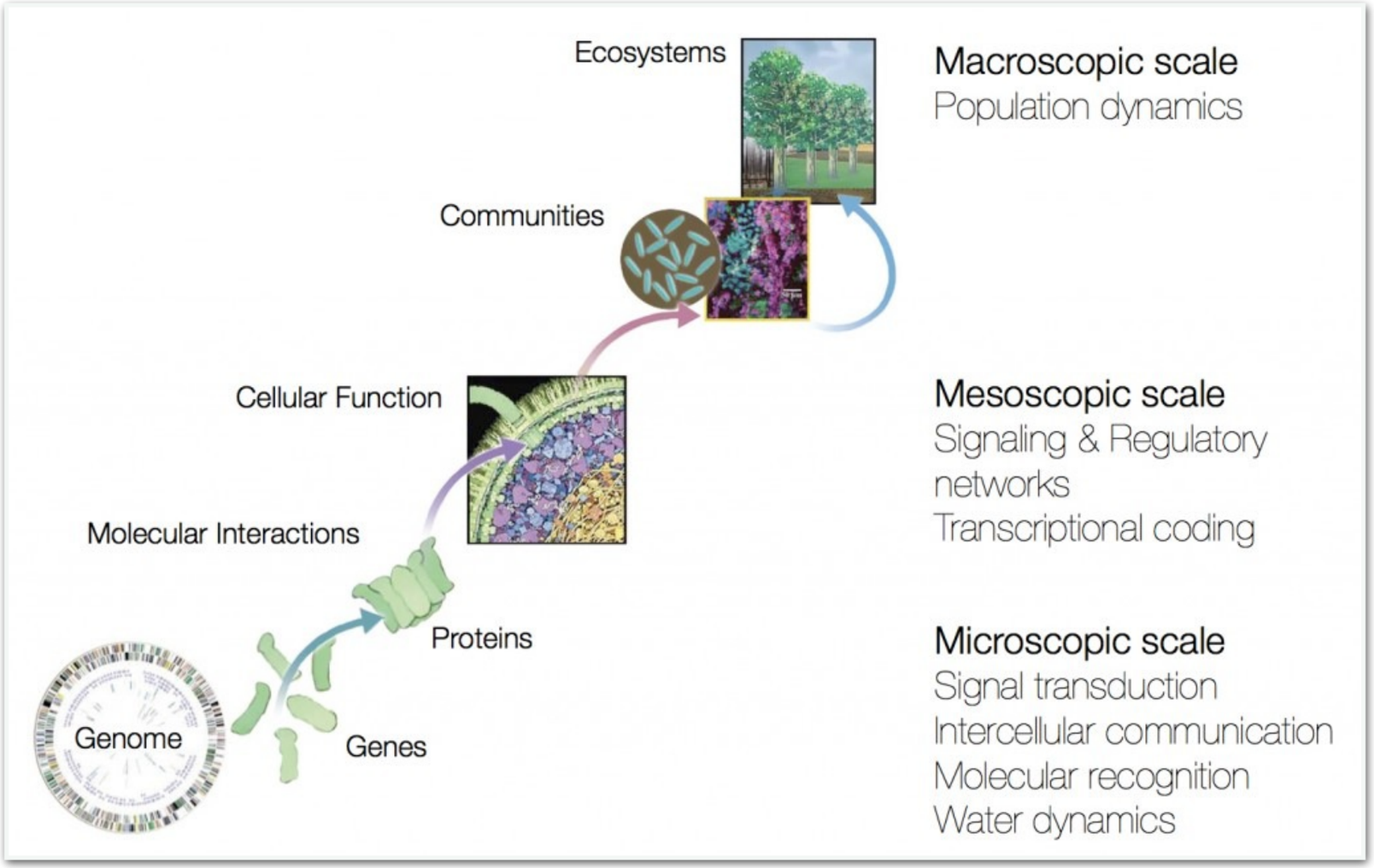
Fernando E. Rosas, PhD

Department of Informatics, University of Sussex

Department of Brain Sciences and Centre for Complexity Science, Imperial College London

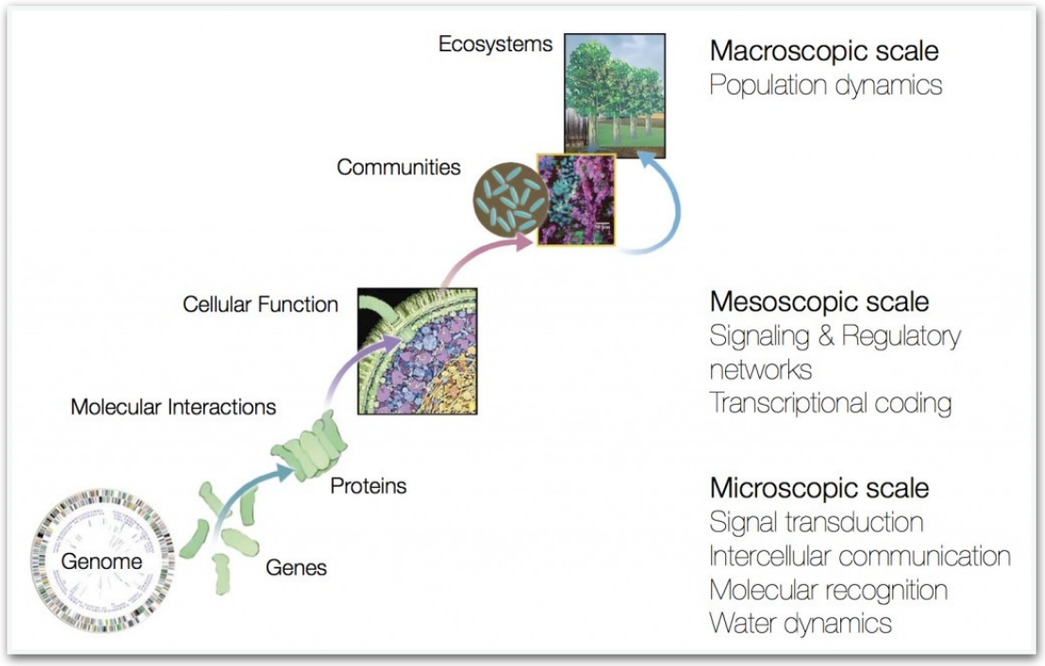
Centre for Eudaimonia and Human Flourishing, University of Oxford

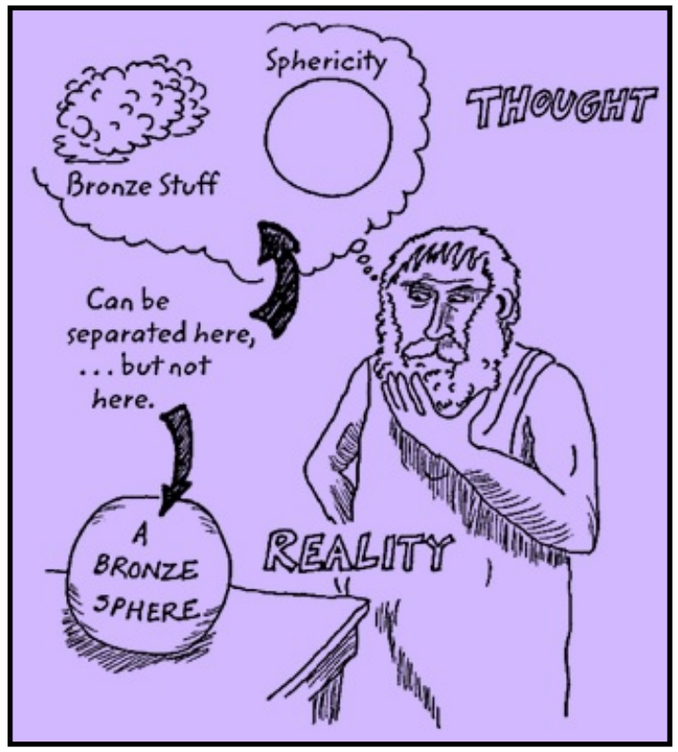
The multi-level nature of our world



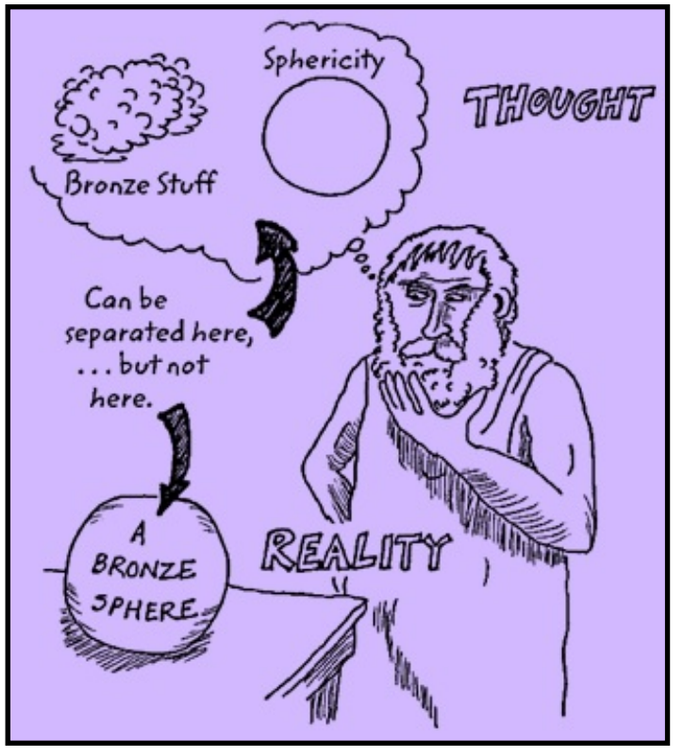
Anderson (1972). More is different: broken symmetry and the nature of the hierarchical structure of science. Science, 177(4047), 393-396.

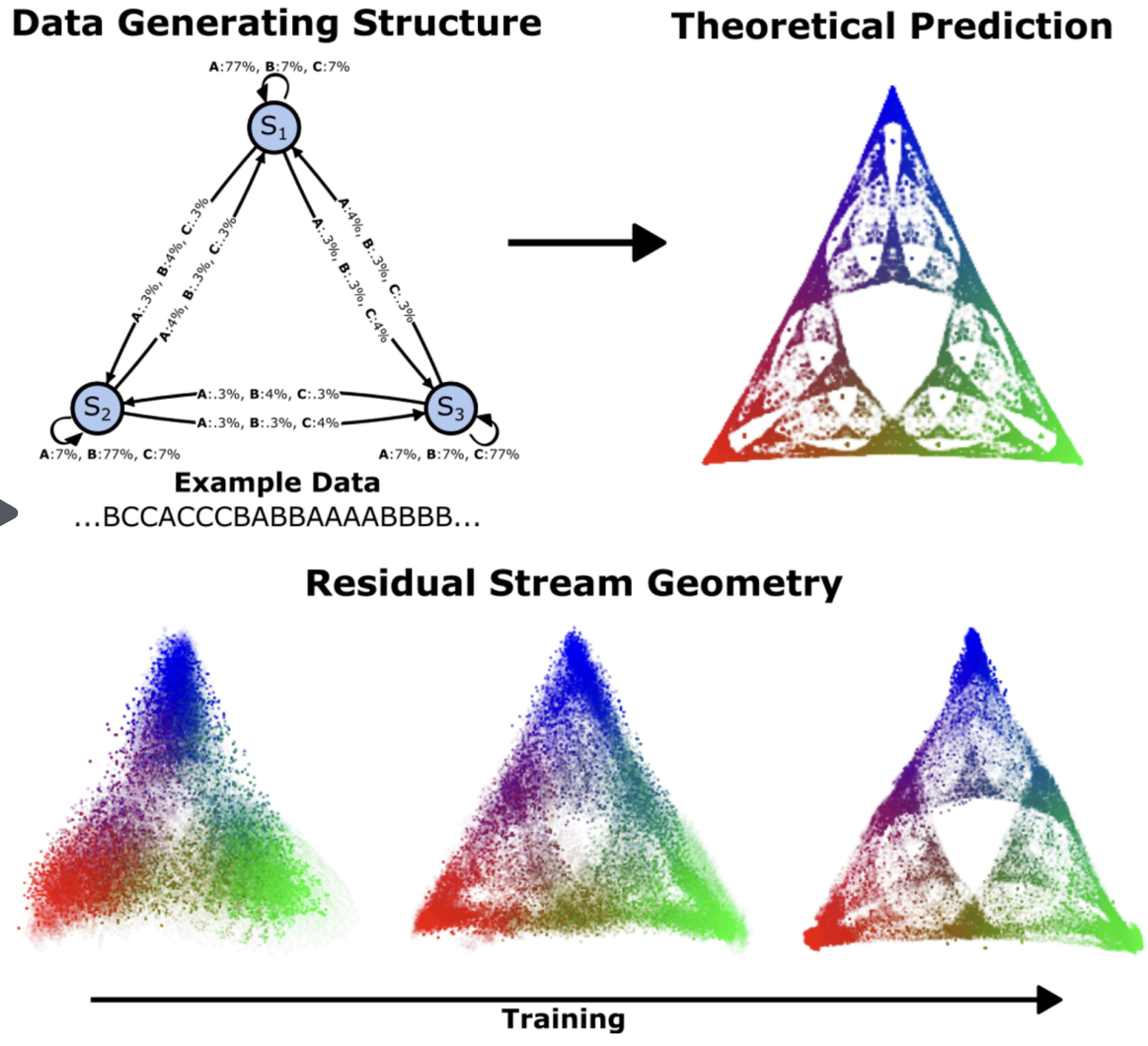
The multi-level nature of our world… is it reflected in our minds?





The multi-level nature of our world… is it reflected in AI's minds?





Shai et al. (2024). Transformers represent belief state geometry in their residual stream. Advances in Neural Information Processing Systems, 37, 75012-75034.

Contents

1. Hierarchical emergence

2. World modelling

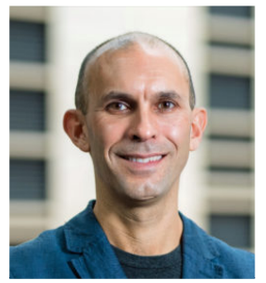
3. Abstractions

4. Ideas to take home

Work developed in synergistic collaboration with :

Bernhard Geiger
Graz

Andrea Luppi
Cambridge



Anil Seth
Sussex

Daniel Polani
Hertfordshire

Michael Gastpar
EPFL

Pedro Mediano
Imperial

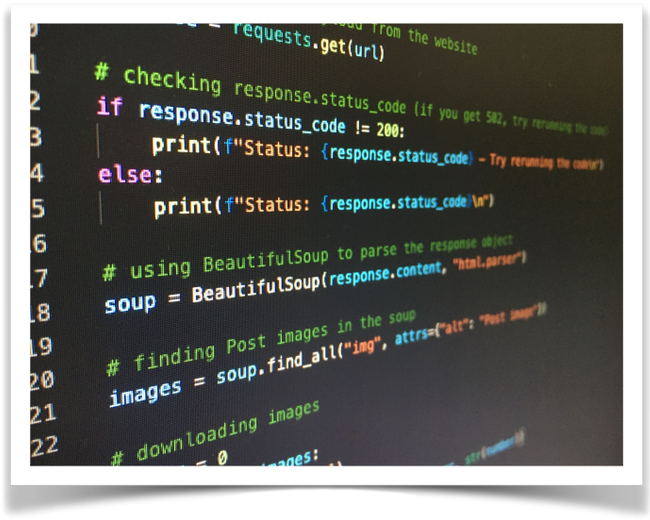
Work supported by the PIBBSS Affiliateship Program.

Rosas et al. (2024). *Software in the natural world: A computational approach to hierarchical emergence*. arXiv preprint arXiv:2402.09090.

Contents
1. Hierarchical emergence
a) Fundamental concepts
b) Results and examples
2. World modelling
3. Abstractions
4. Ideas to take home

How can new laws emerge at coarse-grained levels

The essence of software — having a macroscopic level that is ‘self-contained’.



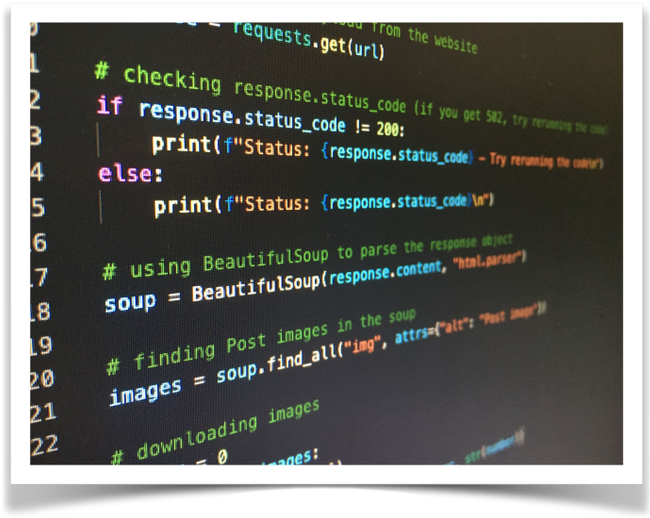

```
a = 2
b = 3
c = a + b
print(c)
```

Key idea: there is a “logic” going on at the macro…

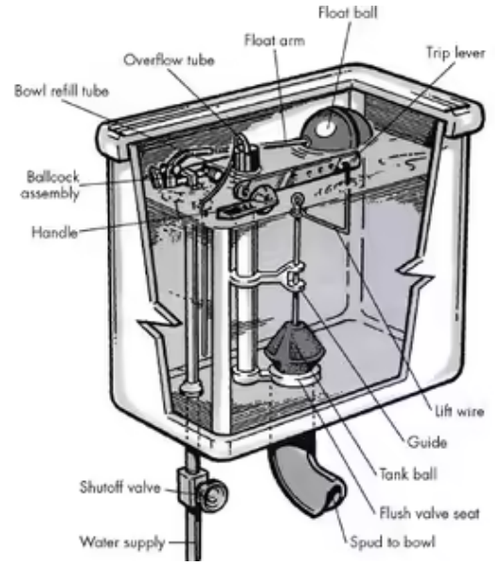
—–> Macro interventions are sufficient to determine macro effects

How can new laws emerge at coarse-grained levels

The essence of software — having a macroscopic level that is ‘self-contained’.




```
a = 2
b = 3
c = a + b
print(c)
```

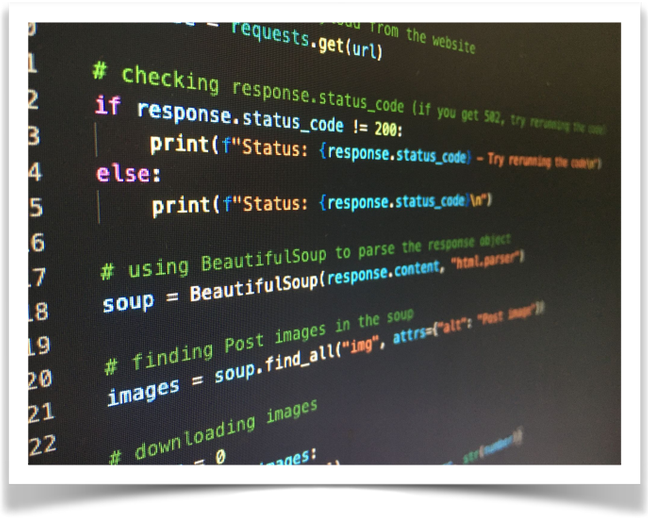


Key idea: there is a “logic” going on at the macro…

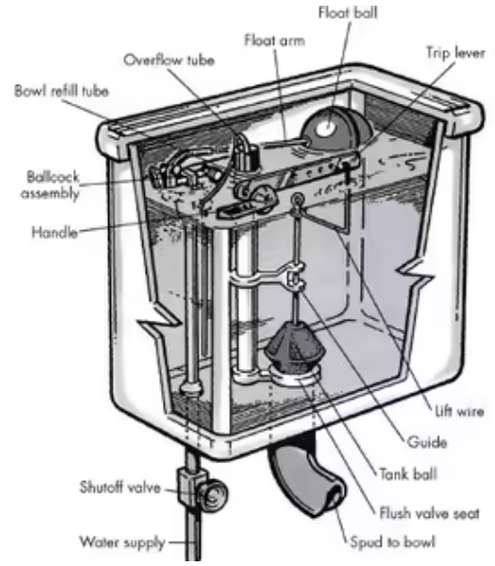
—–> Macro interventions are sufficient to determine macro effects

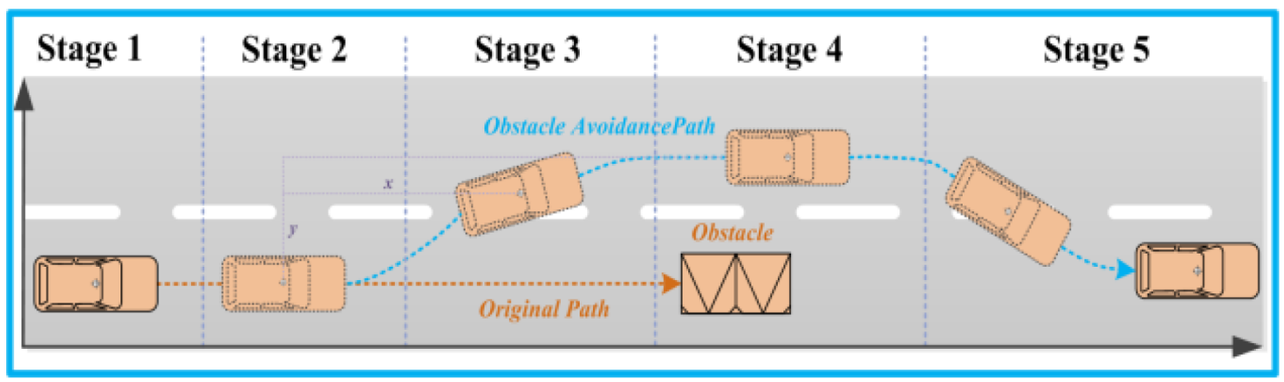
How can new laws emerge at coarse-grained levels

The essence of software — having a macroscopic level that is ‘self-contained’.



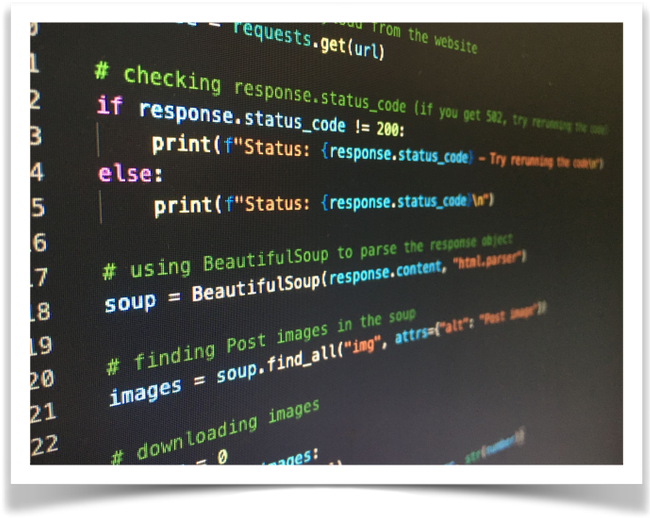

```
a = 2
b = 3
c = a + b
print(c)
```



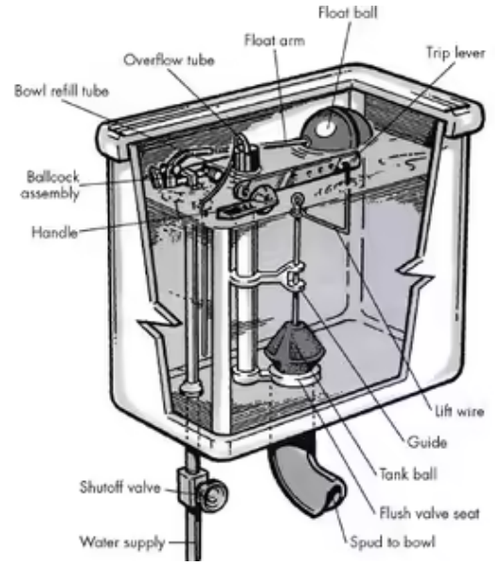


How can new laws emerge at coarse-grained levels

The essence of software — having a macroscopic level that is ‘self-contained’.




```
a = 2
b = 3
c = a + b
print(c)
```



Information closure: Macro-inputs are sufficient to determine macro-outcomes.

Macro level $Z_1, Z_2, \ldots, Z_t$

Micro level $X_1, X_2, \ldots, X_t$

$$Z_t = g(X_1, \ldots, X_t)$$

$$I(Z_1, \ldots, Z_t; Z_{t+1}) = I(X_1, \ldots, X_t; Z_{t+1})$$

equally good prediction from micro or macro

Bertschinger et al. (2006). *Information and closure in systems theory*. In Proceedings of the 7th German Workshop of Artificial Life, pp. 9-21.

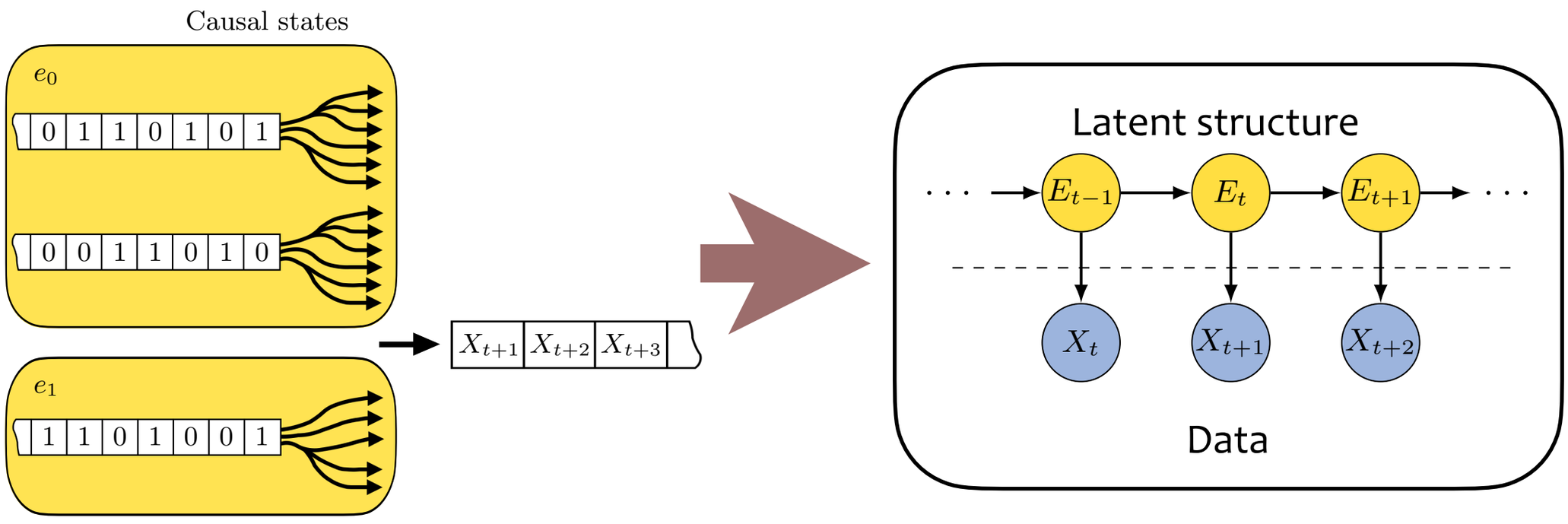
Barnett and Seth (2023). *Dynamical independence: discovering emergent macroscopic processes in complex dynamical systems*. Physical Review E, 108(1), 014304.

Towards a computational understanding of emergence

We investigate emergence using **epsilon-machine**, which give a representation of the data's underlying informational mechanisms.

$$\overleftarrow{\boldsymbol{x}}_t \equiv_\epsilon \overleftarrow{\boldsymbol{x}}'_t \quad \text{iff} \quad p(\vec{\boldsymbol{x}}^L_{t+1}|\overleftarrow{\boldsymbol{x}}_t) = p(\vec{\boldsymbol{x}}^L_{t+1}|\overleftarrow{\boldsymbol{x}}'_t) \quad \forall \vec{\boldsymbol{x}}^L_{t+1}, L \in \mathbb{N}$$

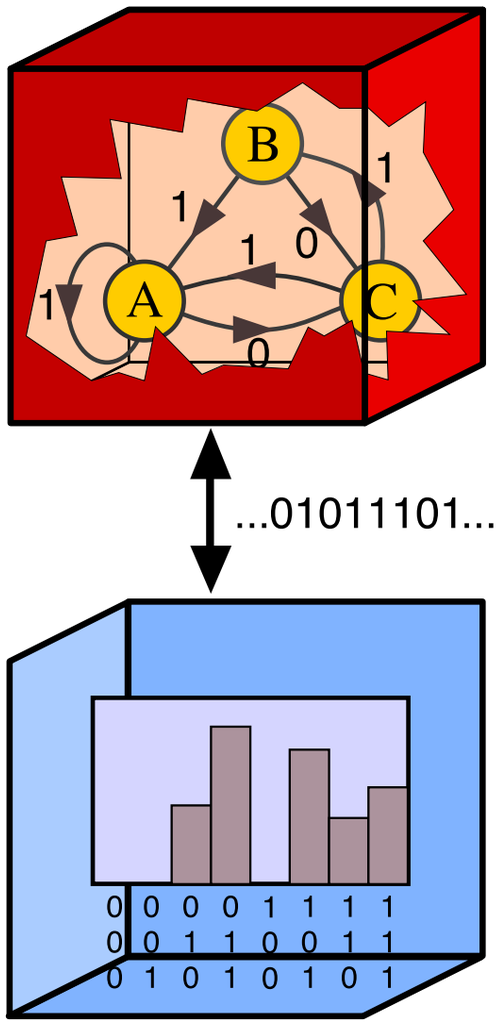
$$T^x_{e,e'} = \mathbb{P}\{E_t = e', X_t = x | E_{t-1} = e\}$$



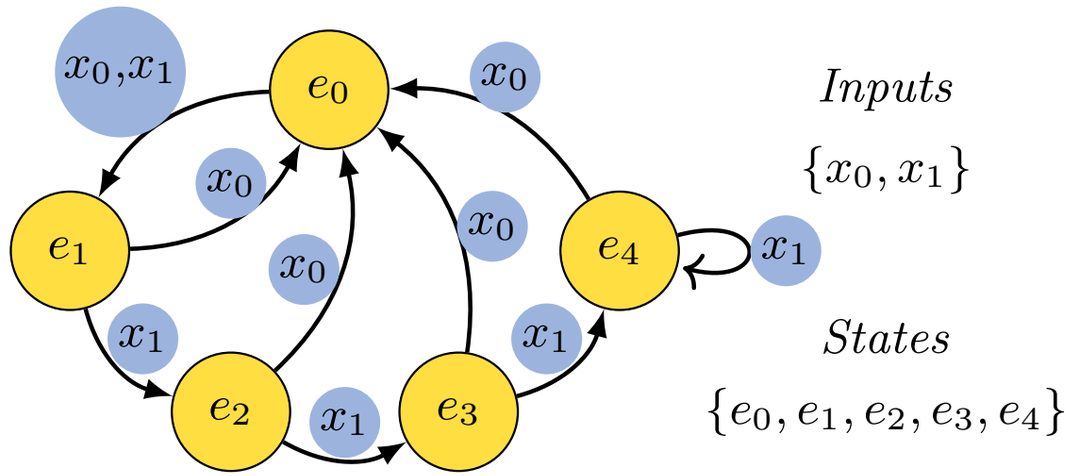
Shalizi, C. R., & Crutchfield, J. P. (2001). Computational mechanics: Pattern and prediction, structure and simplicity. Journal of statistical physics, 104(3), 817-879.

Towards a computational understanding of emergence

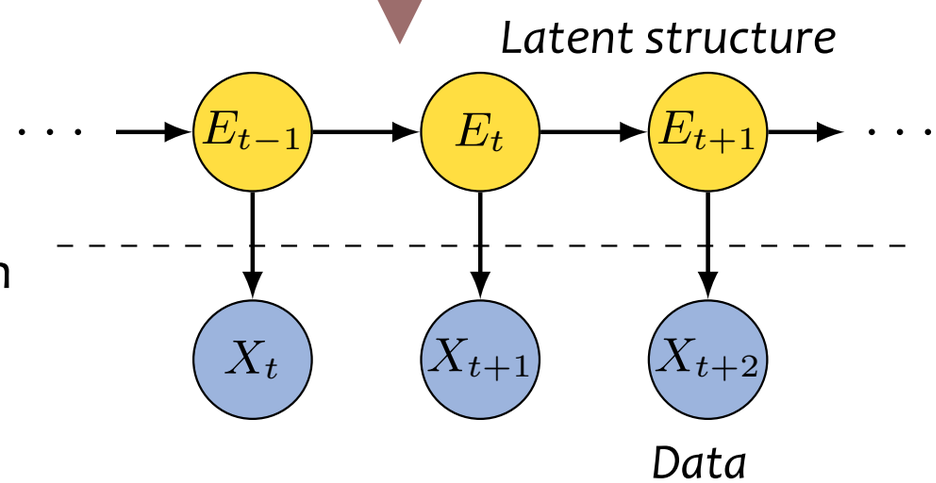
We investigate emergence using **epsilon-machine**, which give a representation of the data's underlying informational mechanisms.



Computational description

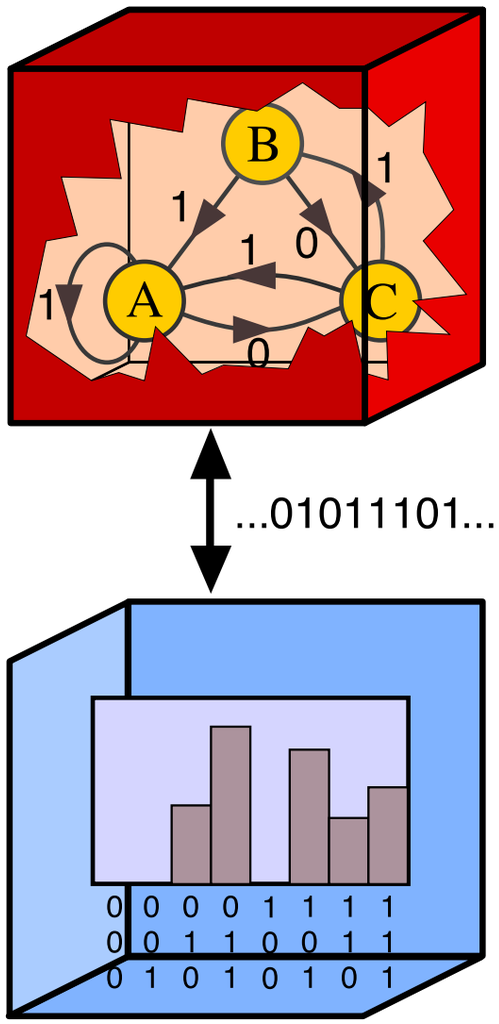


Dynamical description

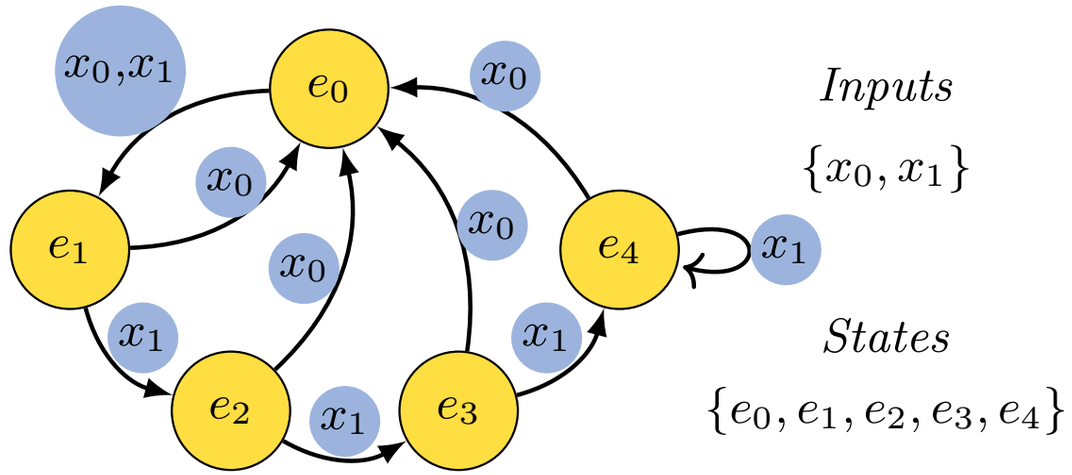


Towards a computational understanding of emergence

We investigate emergence using **epsilon-machine**, which give a representation of the data's underlying informational mechanisms.



Computational description



Automata: $\mathcal{A} = (\mathcal{Q}, \Sigma, \mathcal{O}, \phi, r)$

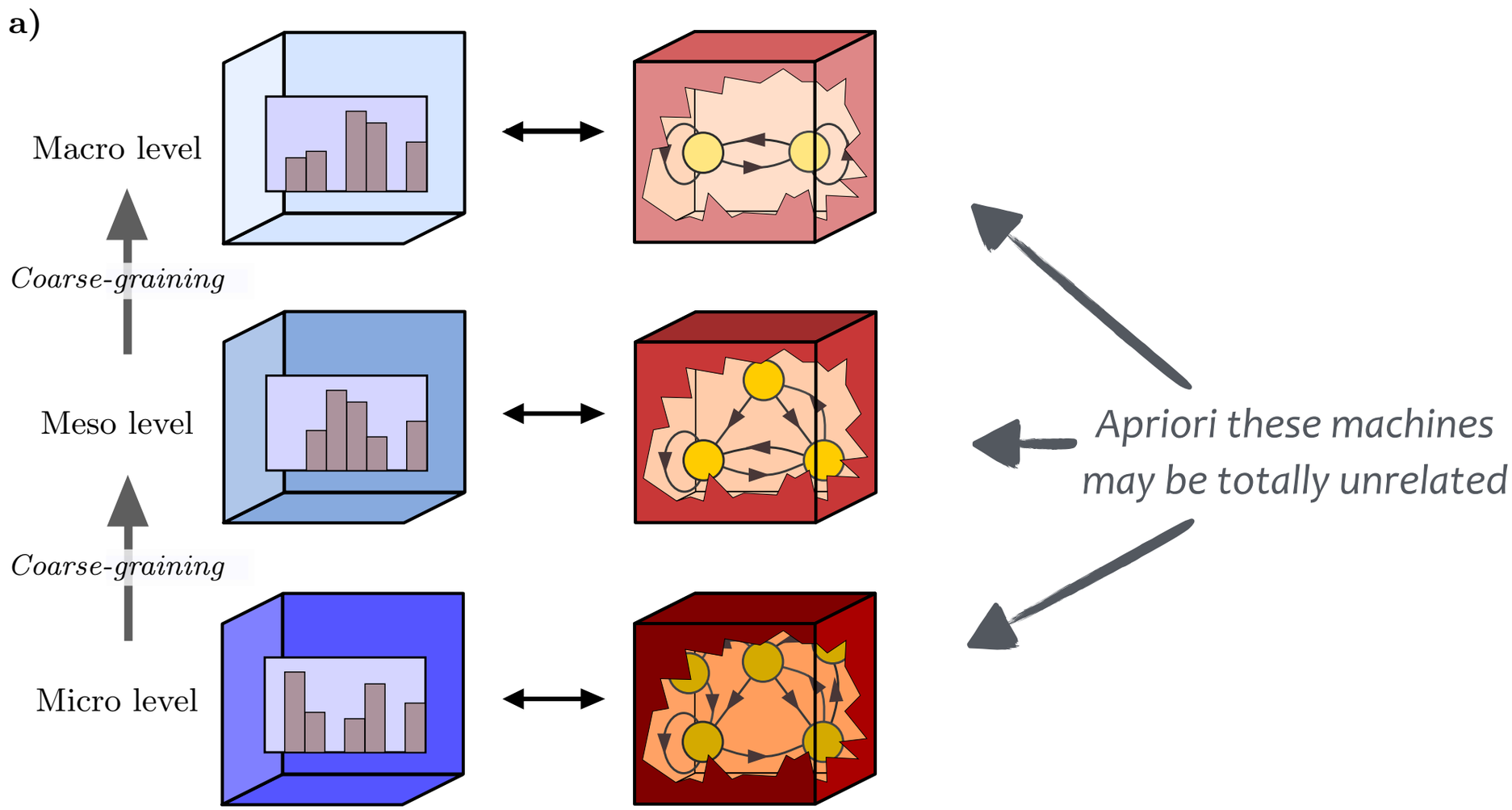
$\mathcal{Q}$ states, Σ inputs, $\mathcal{O}$ outputs

$\phi : \mathcal{Q} \times \Sigma \to \mathcal{Q}$

$r : \Sigma \times \mathcal{Q} \to \mathcal{O}$

Understanding multi-scale computations

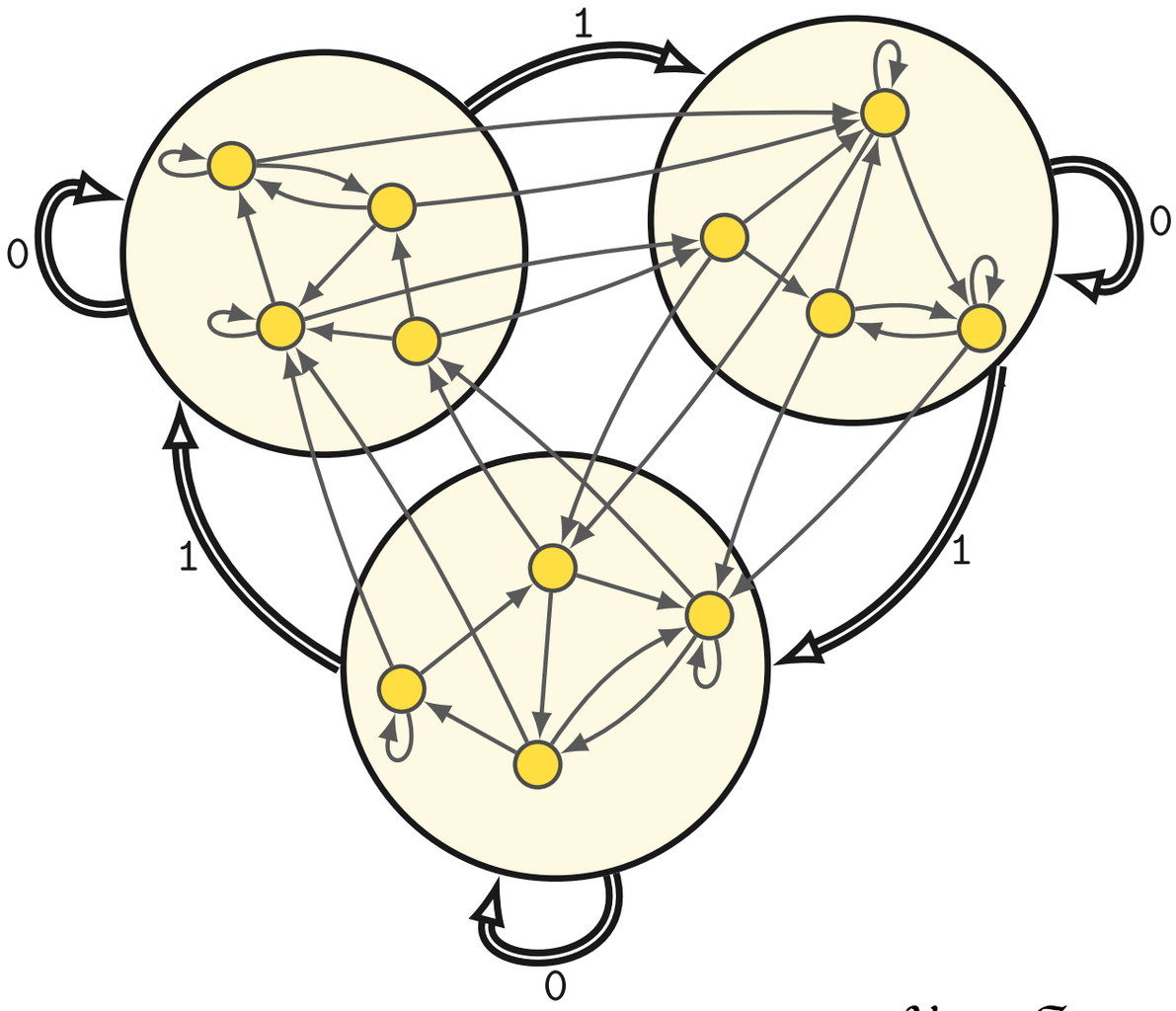
Building epsilon-machines for the data at different scales…



What if they are nested?

Understanding multi-scale computations

A macro process is **computationally closed** if its e-machine is homomorphic to the e-machine of the micro process.



Micro inputs $\mathcal{X} = \{a, b, c\}$

Macro inputs $\mathcal{Z} = \{0, 1\}$

$g : \mathcal{X} \to \mathcal{Z}$

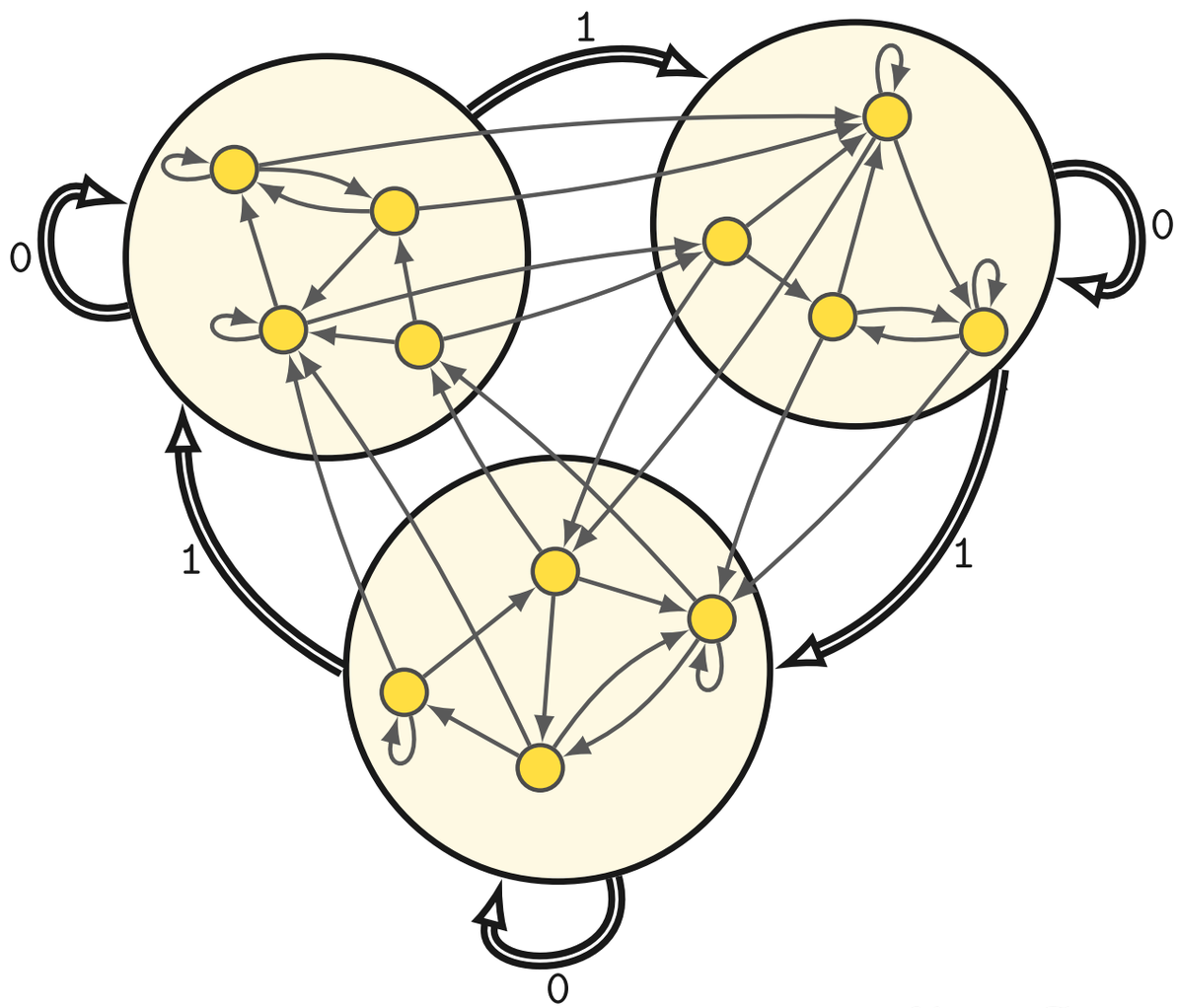
$a \to 0$

$b \to 0$

$c \to 1$

Understanding multi-scale computations

A macro process is **computationally closed** if its e-machine is homomorphic to the e-machine of the micro process.



Micro inputs $\mathcal{X} = \{a, b, c\}$

Macro inputs $\mathcal{Z} = \{0, 1\}$

$g : \mathcal{X} \to \mathcal{Z}$

$a \to 0$

$b \to 0$

$c \to 1$

Automata homomorphism

$$\mathcal{A} = (\mathcal{Q}, \Sigma, \mathcal{O}, \phi, r)$$

$$\mathcal{A}' = (\mathcal{Q}', \Sigma', \mathcal{O}', \phi', r')$$

if there are surjective mappings

$$g : \Sigma \to \Sigma' \qquad h : \mathcal{Q} \to \mathcal{Q}'$$

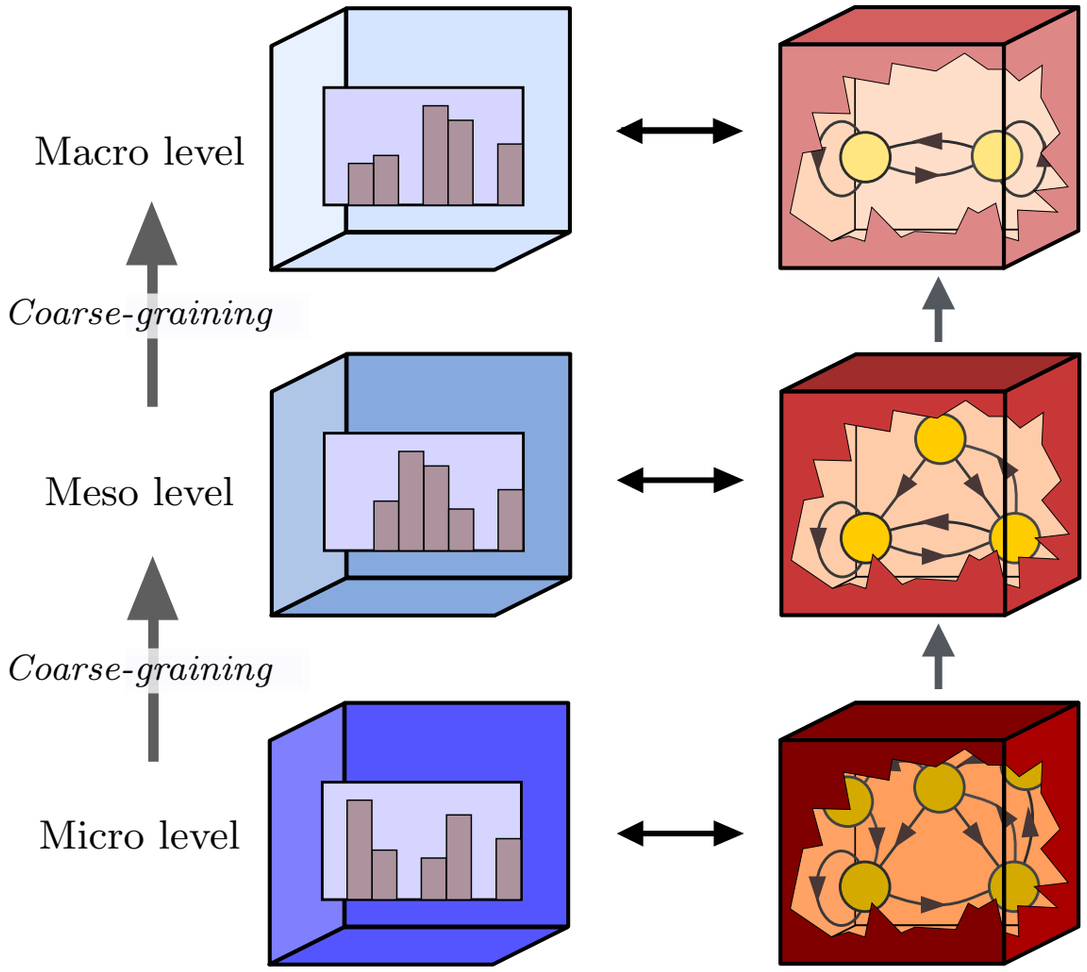
such that

$$h\big(\phi(s, q)\big) = \phi'\big(g(s), h(q)\big)$$

$$r(s', q') = \sup_{g(s)=s', h(q)=q'} r(s, q)$$

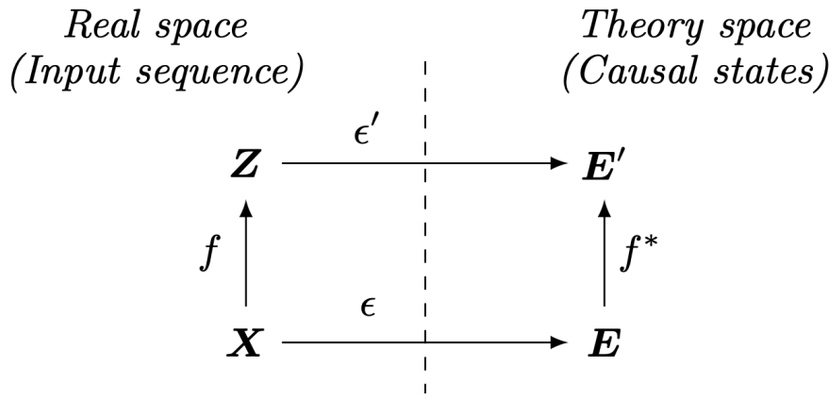
Understanding multi-scale computations

A macro process is **computationally closed** if its e-machine is homomorphic to the e-machine of the micro process.



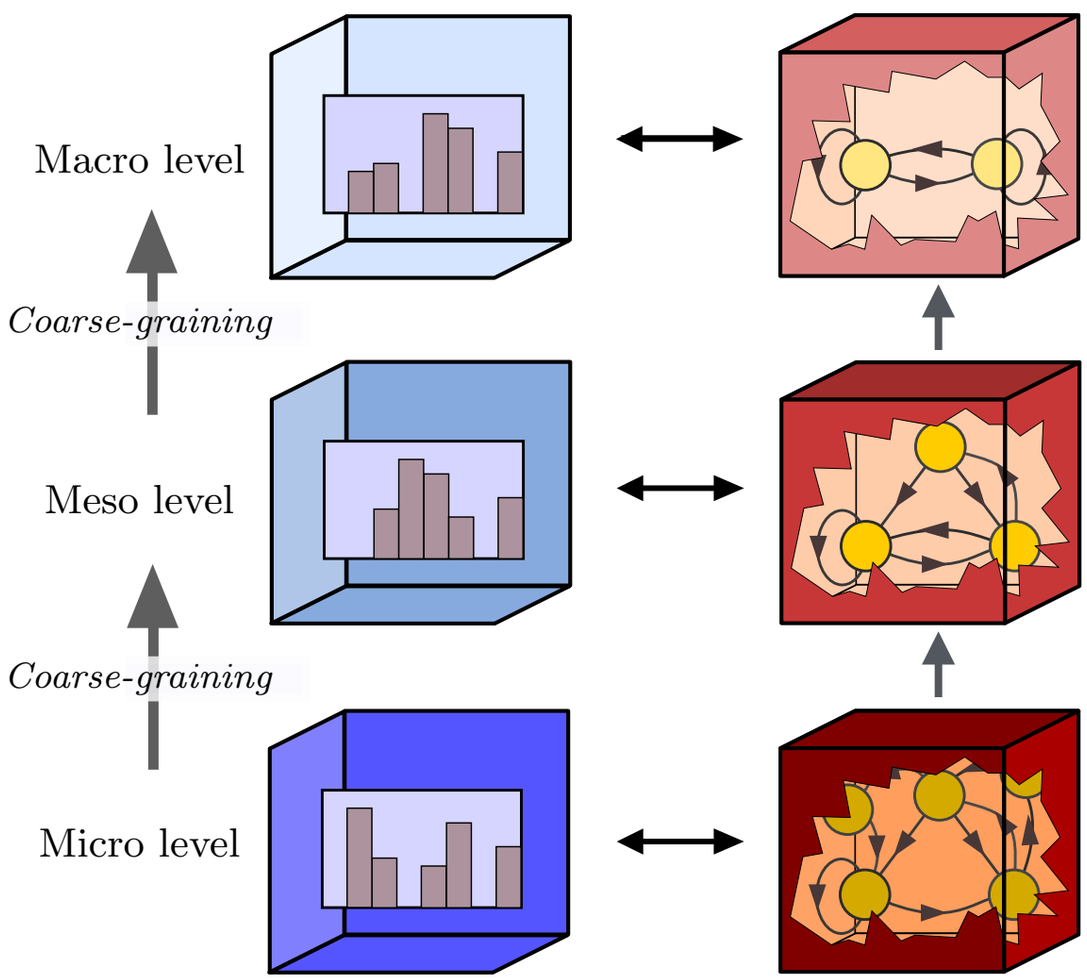
Key idea:

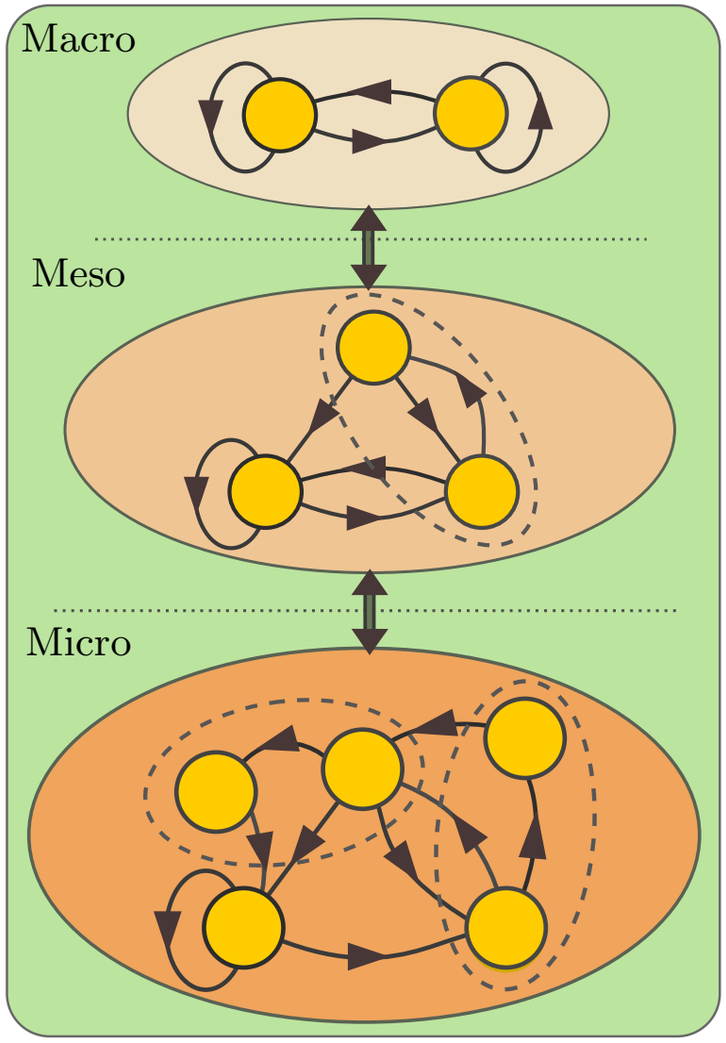
Computational closure guarantees the equivalence of coarse-graining data or 'theories'



Understanding multi-scale computations

A macro process is **computationally closed** if its e-machine is homomorphic to the e-machine of the micro process.





Example: Random walk over a modular network

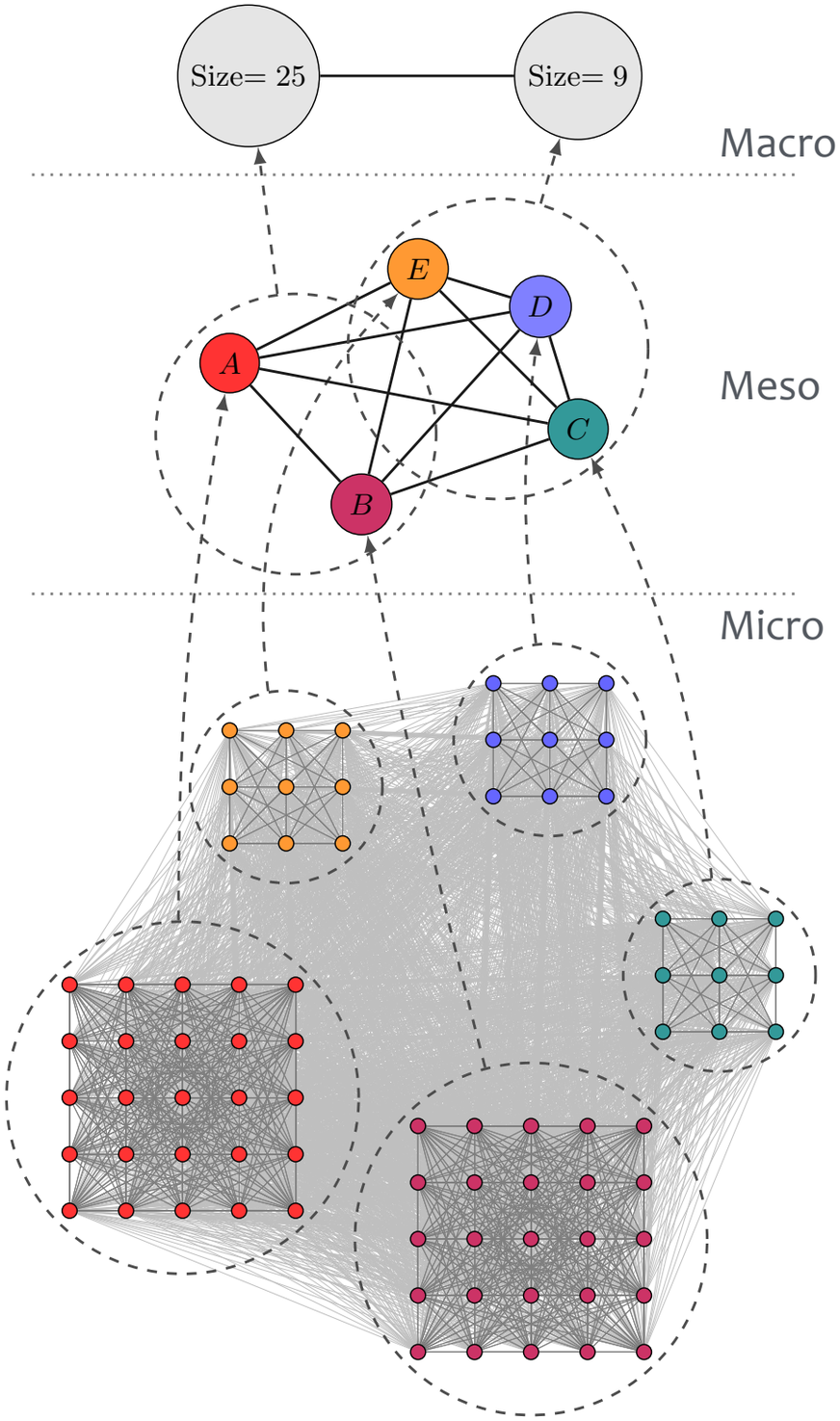
System is described by

- Probability of walking: q
- Within cluster connectivity: r_{w}
- Between cluster connectivity: r_{b}

Three levels of description:

- Micro – which node.
- Meso – which community.
- Macro – which size of community.

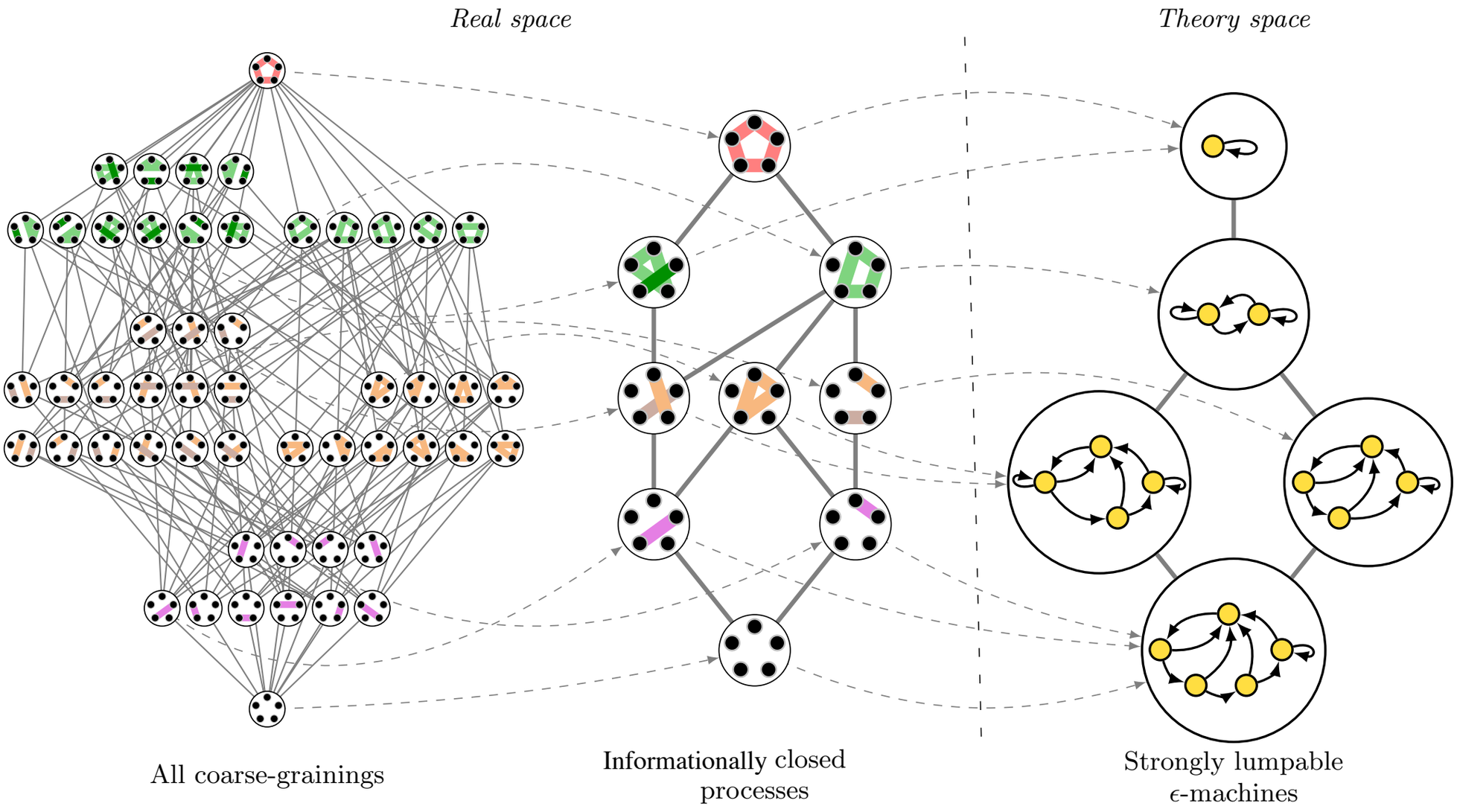
The computation that leads to the next state can be described hierarchically!



Contents
1. Hierarchical emergence
a) Fundamental concepts
b) Results and examples
2. World modelling
3. Abstractions
4. Ideas to take home

Theory: The lattice of emergent processes

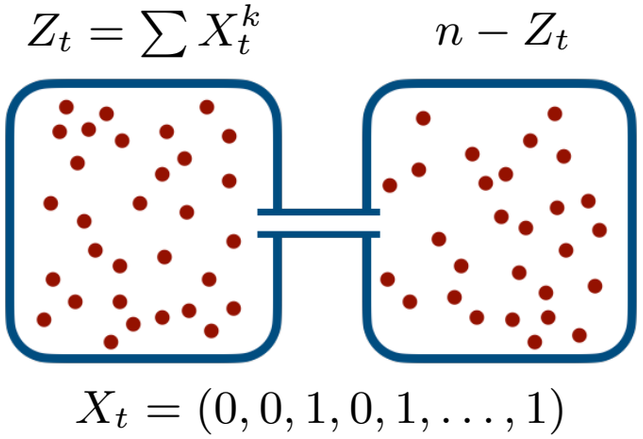
Theorem:
Information/causal closure implies computational closure

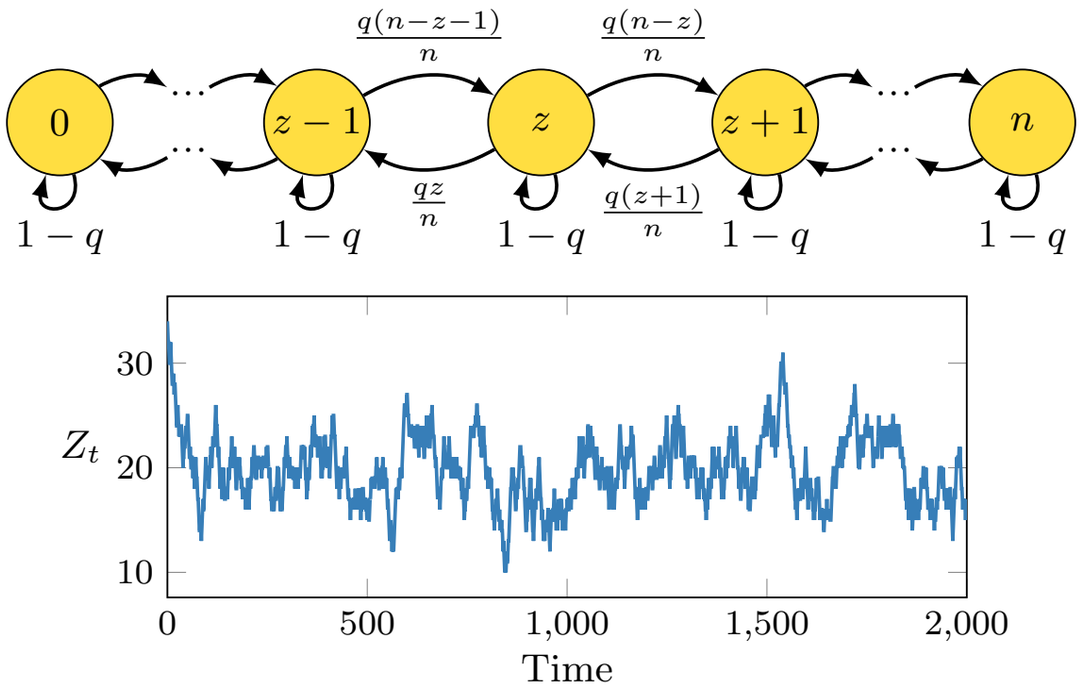


Rosas et al. (2024). *Software in the natural world: A computational approach to hierarchical emergence*. arXiv preprint arXiv:2402.09090.

Thermodynamic and neural examples

- **Ehrenfest diffusion model**
  n particles diffusing with probability *q*





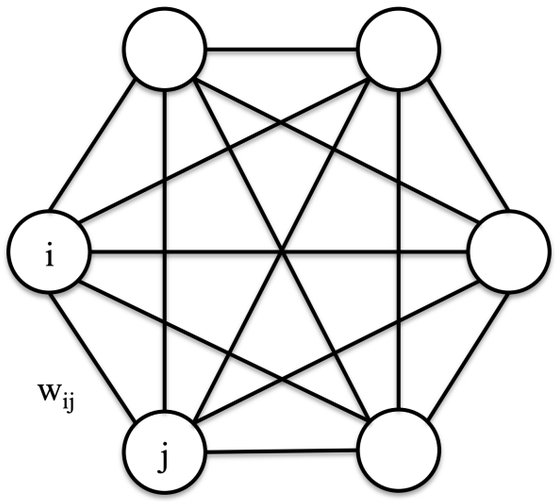
The dynamics of the number of particles in left chamber is computationally closed

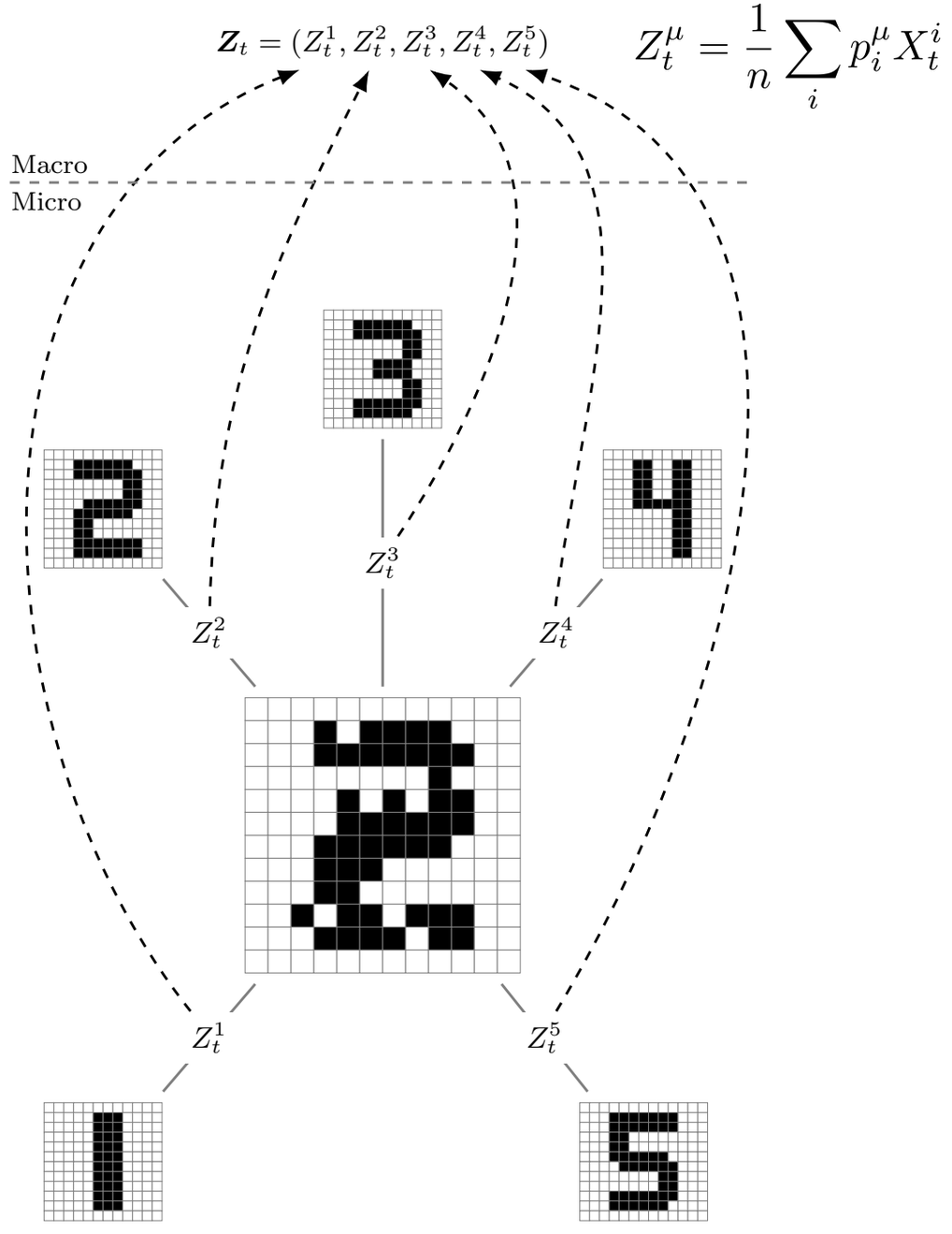
Rosas et al. (2024). *Software in the natural world: A computational approach to hierarchical emergence*. arXiv preprint arXiv:2402.09090.

Thermodynamic and neural examples

- **Hopfield networks**
  RNN that models memory retrieval

$$w_{i,j} = \frac{1}{n}\sum_{\mu=1}^{m} p_i^\mu p_j^\mu$$





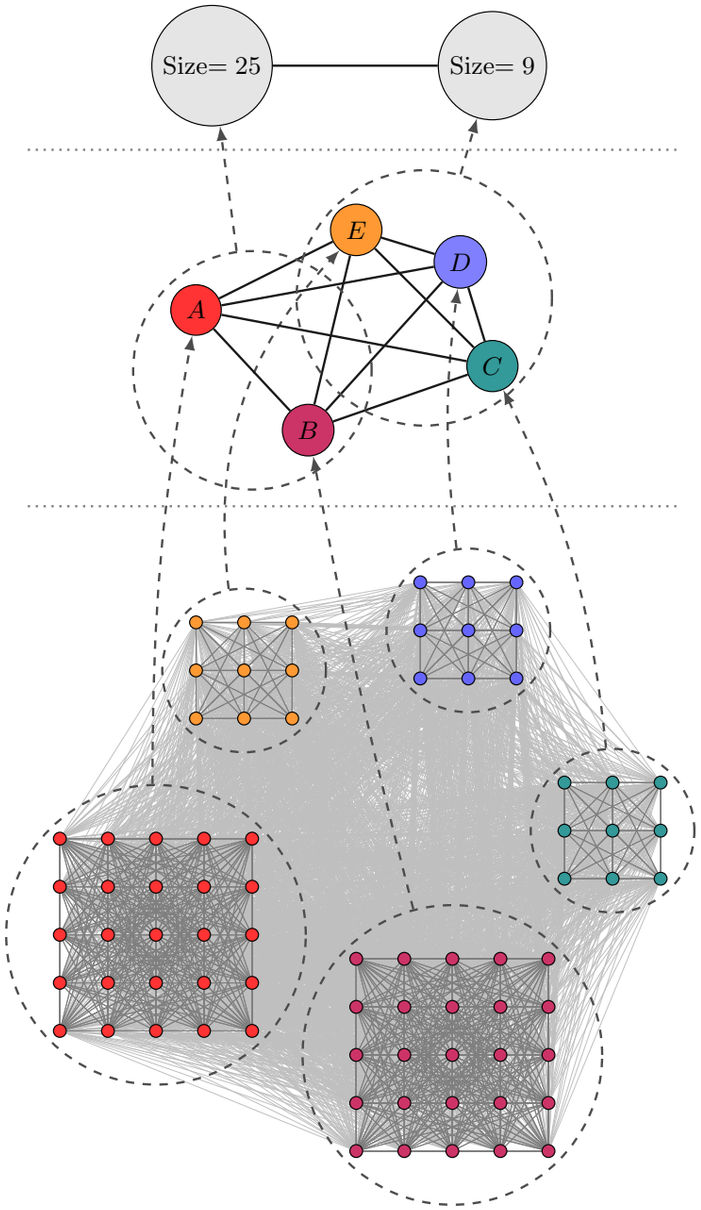
The dynamics of the dot products determine which memory is retrieved

Application: Natural abstractions in transformers (ongoing)

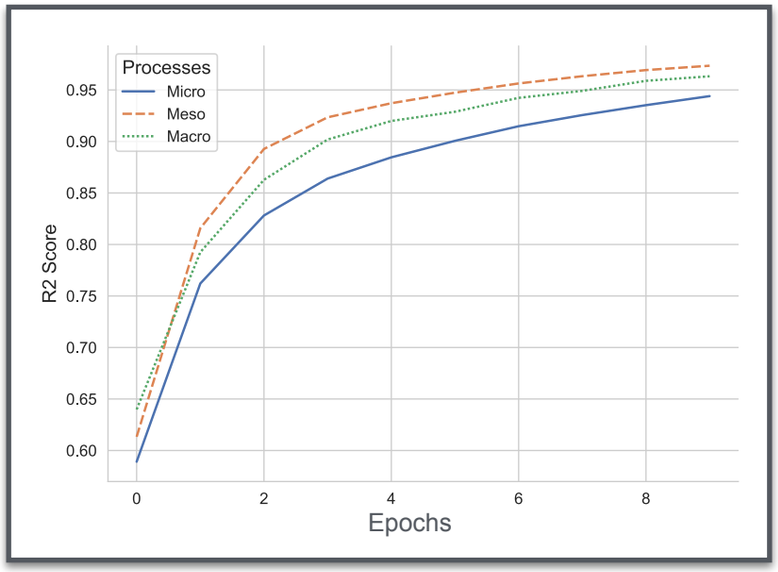
Scenario:

- Generate data from a multi-scale process.
- Use it to train transformers of various sizes.

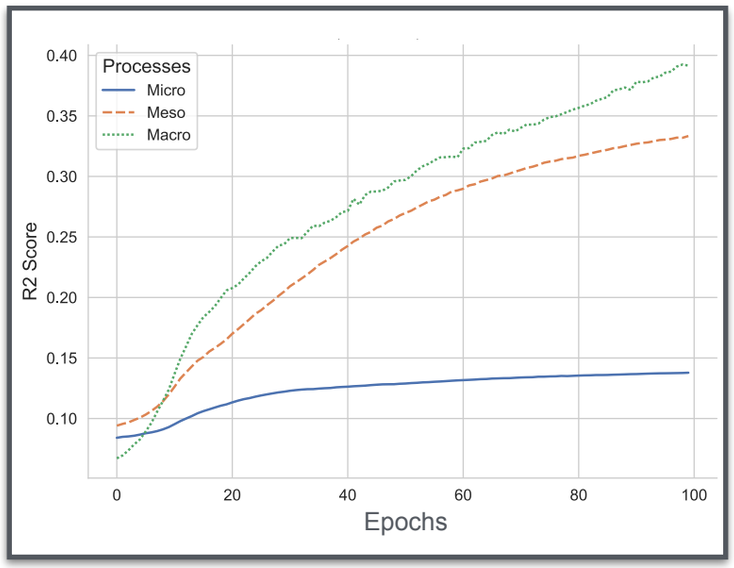
Conjecture — Small transformers will naturally find computationally closed levels…



Large transformer



Small transformer

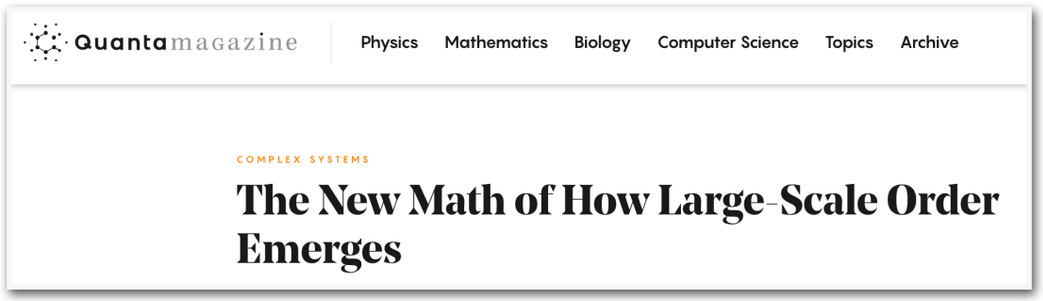


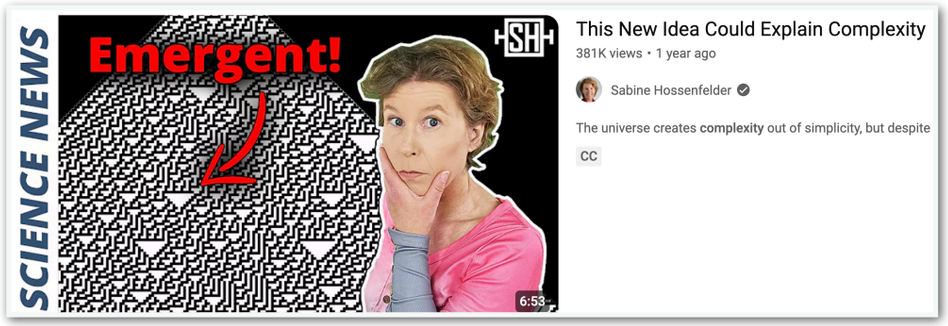
Transformer seem to naturally find computationally closed levels

Project supported by PIBBSS

Interim summary

- ✦ Hierarchical emergence highlight scales that are computationally autonomous.
- ✦ Computationally closed levels are amenable to efficient controllability and simulations.
- ✦ Hierarchical emergence is common in statistical mechanics and computational neuroscience models.
- ✦ Transformers seem to naturally identify computationally closed levels.





Contents

1. Hierarchical emergence

2. World modelling

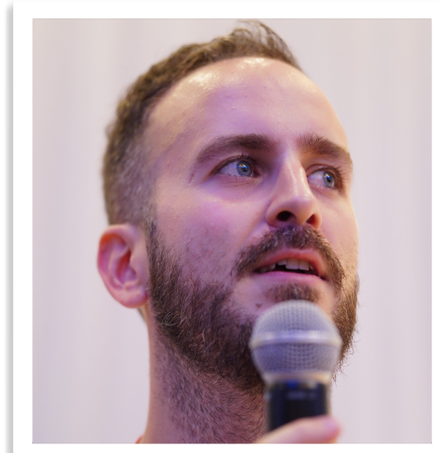
3. Abstractions

4. Ideas to take home

Work developed in synergistic collaboration with :

Alexander Boyd

(University of Sussex, BITS)



Manuel Baltieri

(Araya)

Work supported by the ARIA's *Safeguraded AI* and PIBBSS affiliateship programs.

Rosas, Boyd and Baltieri, "AI in a vat: Fundamental limits of efficient world modelling for agent sandboxing and interpretability," in Reinforcement Learning Journal, vol. 6, pp. 2844–2881, Aug. 2025.

Contents

1. Hierarchical emergence

2. World modelling

a) Setting and formal foundations

b) Minimal world models

c) Interpretable world models

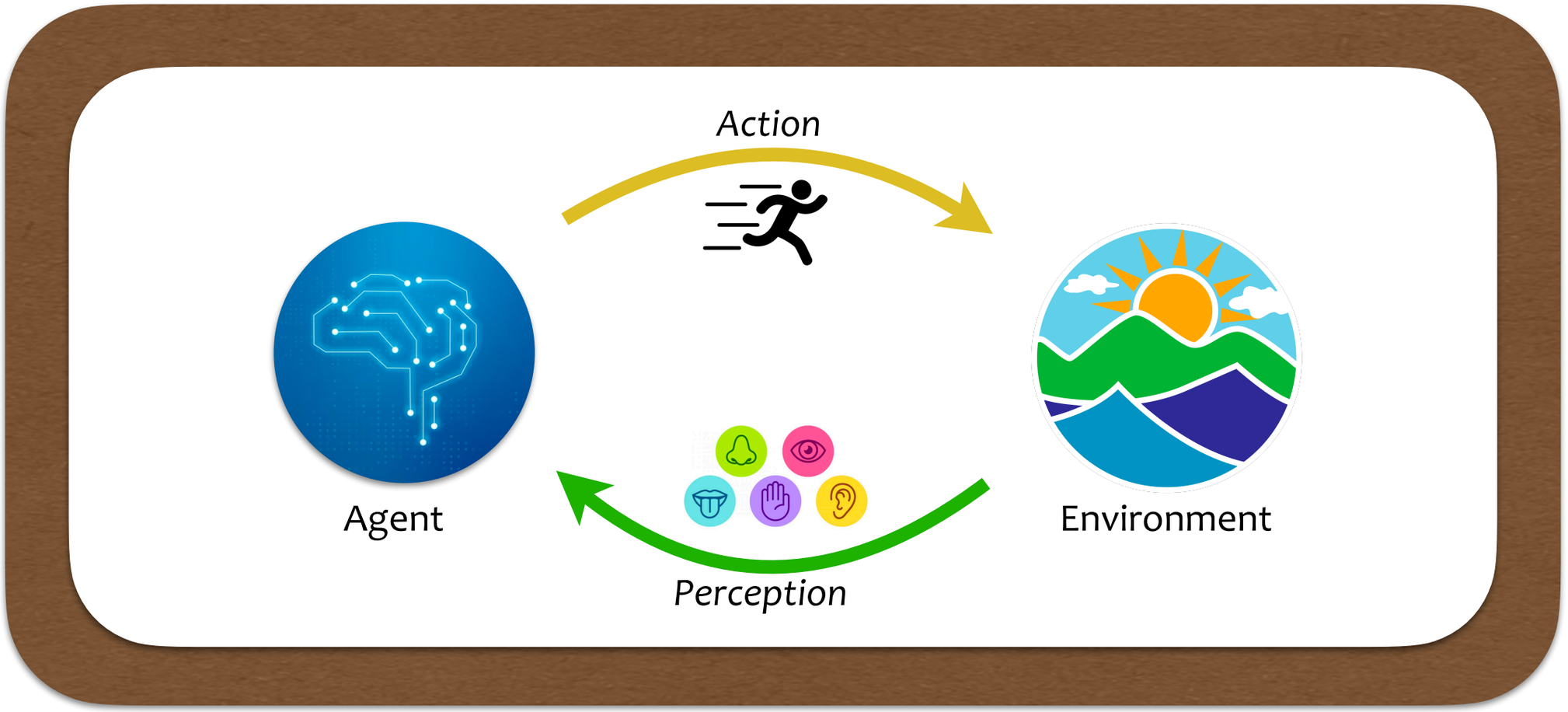
3. Abstractions

4. Ideas to take home

AI sandboxing

Scenario:

- An AI system needs testing before deployment
- There is a probabilistic model of the world available for sandboxing



AI sandboxing

Scenario:

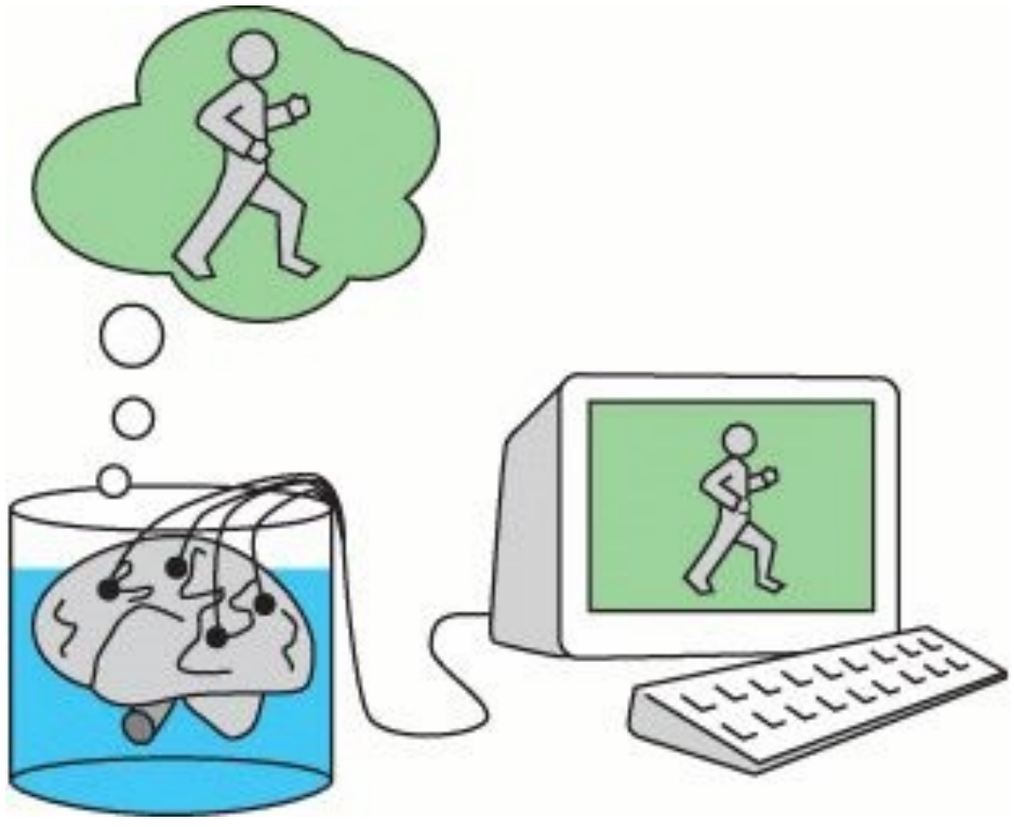
- An AI system needs testing before deployment
- There is a probabilistic model of the world available for sandboxing

Reasons to do better:

- The model may be too detailed (e.g. a physical model of the whole planet)
- Agent may not “see” some resolution (e.g. classical agents cannot see quantum)
- The agent may only interact with a corner of the world (e.g. a bacteria)

Thought experiment: Brain in a Vat

How can one distinguish between reality and a simulation?



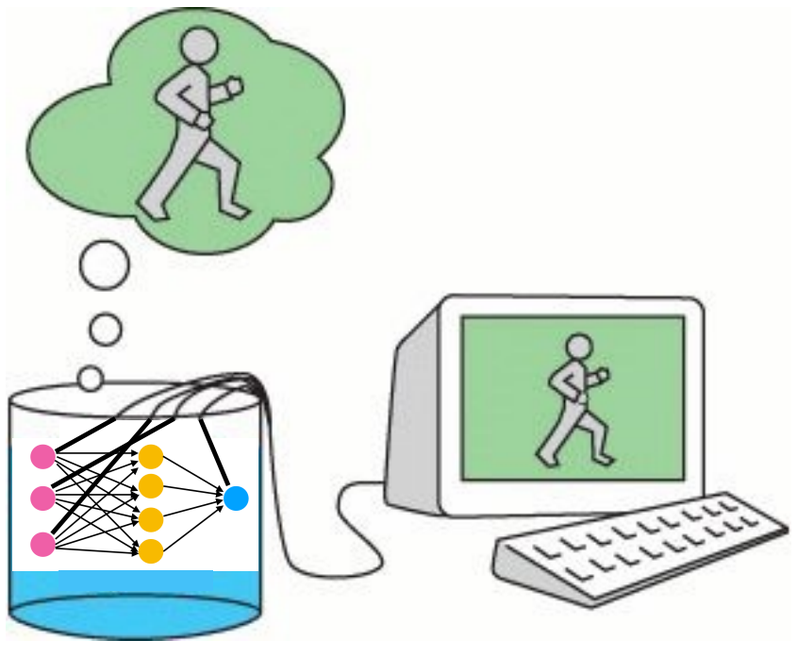
- Plato's allegory of the cave, *The Republic*, (~375 BC)
- Zhuangzi's butterfly dream (circa 476-221 BC)
- Decartes' evil demon, *Meditations* (1641)
- Hilary Putnam, *Brains in a vat* (1981)
- *The Matrix* (1999)

Thought experiment: Brain in a Vat

How can one distinguish between reality and a simulation?

Key insight ("AI in a vat" approach):

—> Forget about the world, and focus on how inputs turn into outputs (the interface)

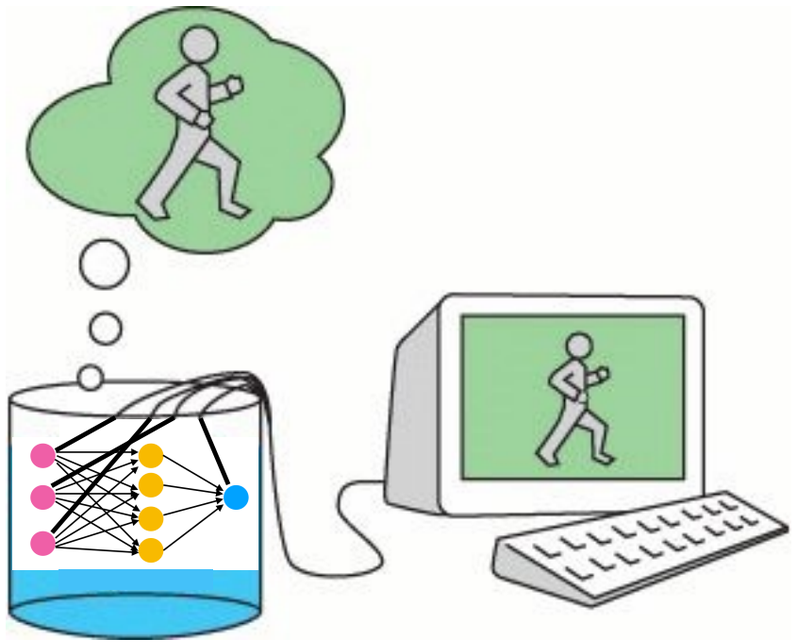


Thought experiment: Brain in a Vat

How can one distinguish between reality and a simulation?

Key insight ("AI in a vat" approach):

—> Forget about the world, and focus on how inputs turn into outputs (the interface)

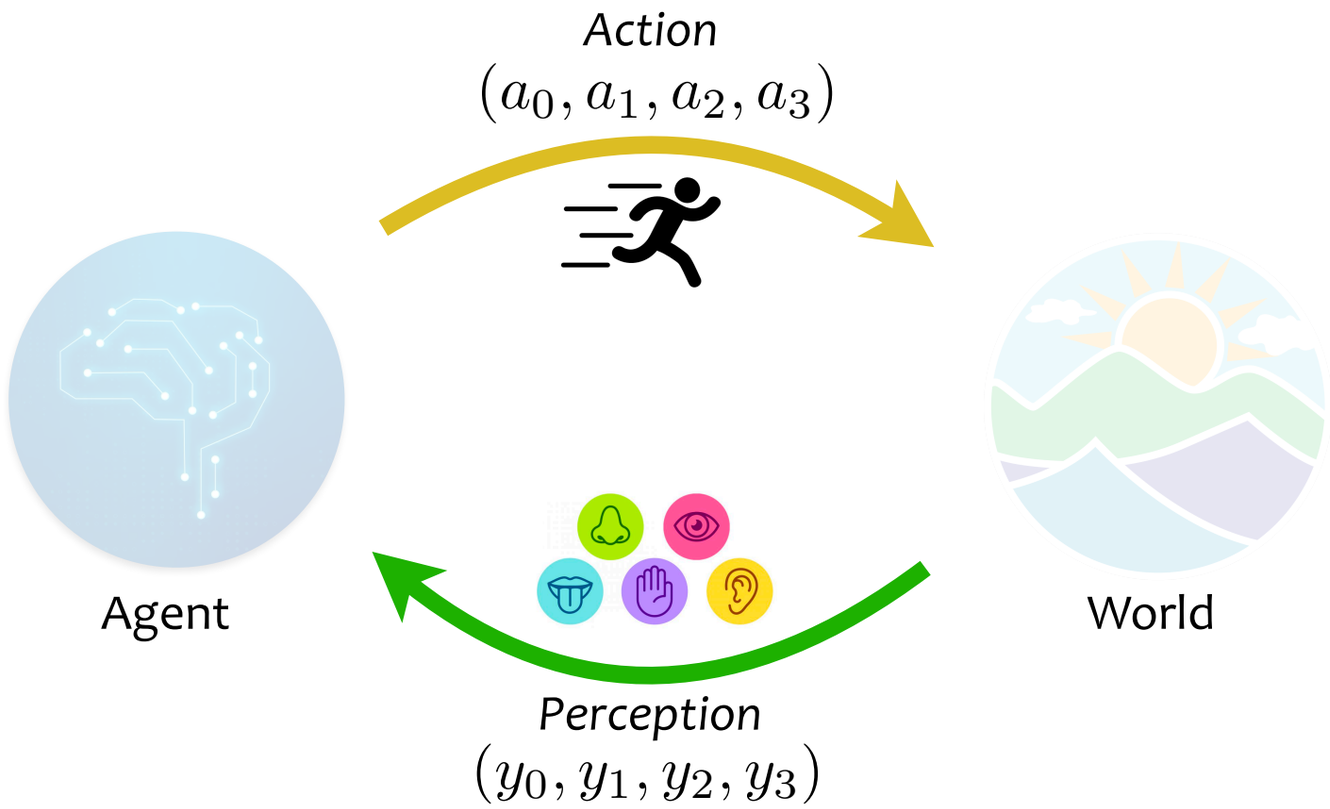


How can we best simulate an interface? (*computational goal*)

Can we characterise what can be learned through it? (*epistemic goal*)

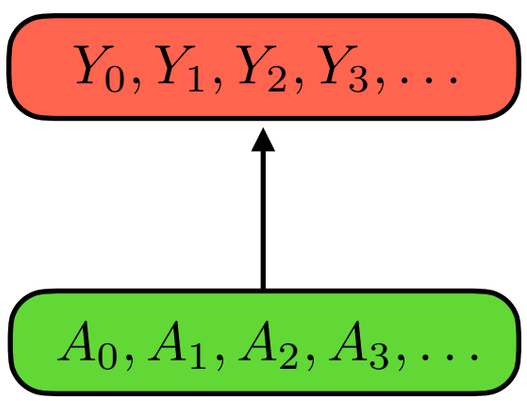
Formal foundations

The **interface of an agent** $\mathcal{I}(\boldsymbol{Y}|\boldsymbol{A})$ is a collection of conditional distributions $p(\boldsymbol{y}_{:t}|\boldsymbol{a}_{:})$ turning sequences of actions into outcomes, satisfying $p(\boldsymbol{y}_{:t}|\boldsymbol{a}_{:}) = p(\boldsymbol{y}_{:t}|\boldsymbol{a}_{:t})$.



Formal foundations

The **interface of an agent** $\mathcal{I}(\boldsymbol{Y}|\boldsymbol{A})$ is a collection of conditional distributions $p(\boldsymbol{y}_{:t}|\boldsymbol{a}_{:})$ turning sequences of actions into outcomes, satisfying $p(\boldsymbol{y}_{:t}|\boldsymbol{a}_{:}) = p(\boldsymbol{y}_{:t}|\boldsymbol{a}_{:t})$.

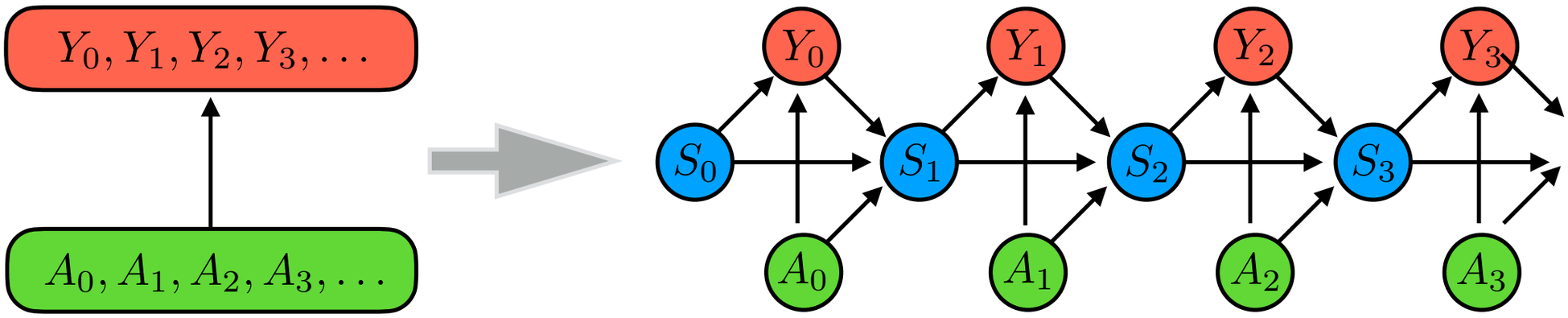


Formal foundations

The **interface of an agent** $\mathcal{I}(\boldsymbol{Y}|\boldsymbol{A})$ is a collection of conditional distributions $p(\boldsymbol{y}_{:t}|\boldsymbol{a}_{:})$ turning sequences of actions into outcomes, satisfying $p(\boldsymbol{y}_{:t}|\boldsymbol{a}_{:}) = p(\boldsymbol{y}_{:t}|\boldsymbol{a}_{:t})$.

A **Markov world model** for a given interface $\mathcal{I}(\boldsymbol{Y}|\boldsymbol{A})$ is another collection of distributions $p(\boldsymbol{s}_{:t}|\boldsymbol{h}_{:})$ corresponding to an auxiliary process leading to the following factorisation:

$$p(\boldsymbol{y}_{:t}\boldsymbol{s}_{:t+1}|\boldsymbol{a}_{:}) = p(s_0)\prod_{\tau=0}^{t}\kappa_\tau(y_\tau, s_{\tau+1}|a_\tau, s_t)$$



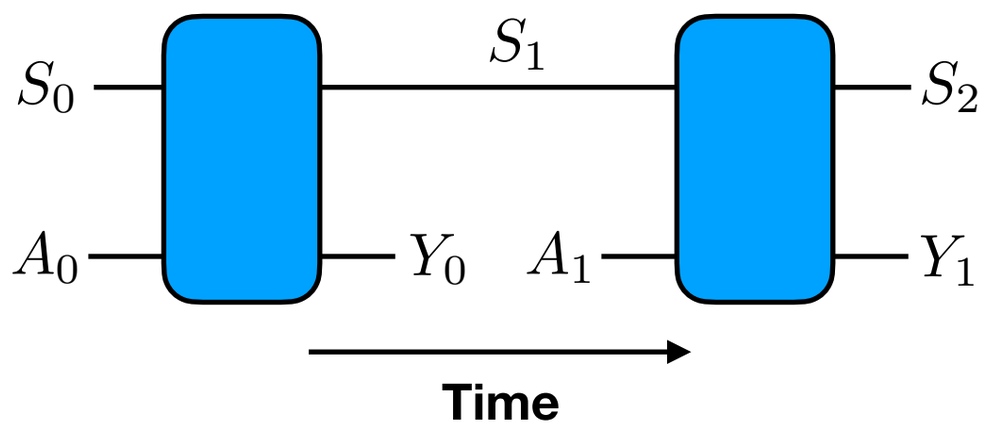
Formal foundations

The **interface of an agent** $\mathcal{I}(\boldsymbol{Y}|\boldsymbol{A})$ is a collection of conditional distributions $p(\boldsymbol{y}_{:t}|\boldsymbol{a}_{:})$ turning sequences of actions into outcomes, satisfying $p(\boldsymbol{y}_{:t}|\boldsymbol{a}_{:}) = p(\boldsymbol{y}_{:t}|\boldsymbol{a}_{:t})$.

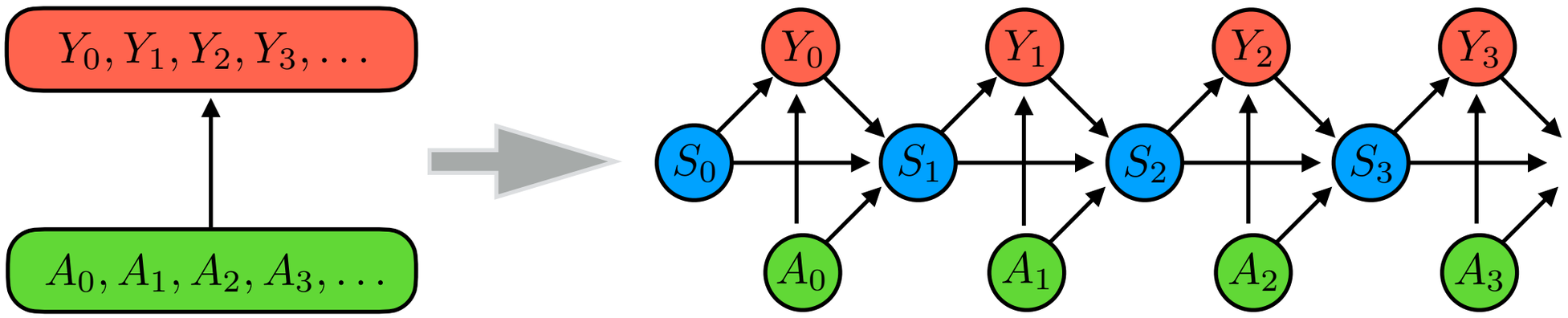
A **Markov world model** for a given interface $\mathcal{I}(\boldsymbol{Y}|\boldsymbol{A})$ is another collection of distributions $p(\boldsymbol{s}_{:t}|\boldsymbol{h}_{:})$ corresponding to an auxiliary process leading to the following factorisation:

$$p(\boldsymbol{y}_{:t}\boldsymbol{s}_{:t+1}|\boldsymbol{a}_{:}) = p(s_0)\prod_{\tau=0}^{t}\kappa_\tau(y_\tau, s_{\tau+1}|a_\tau, s_t)$$

Markov world models correspond to **transducer:** stochastic automata $(\mathcal{S}, \mathcal{A}, \mathcal{Y}, \kappa, p)$ with states in $\mathcal{S}$, inputs $\mathcal{A}$, outputs $\mathcal{Y}$, and a Markov kernel $\kappa(y, \tilde{s}|a, s)$.



Properties of transducers

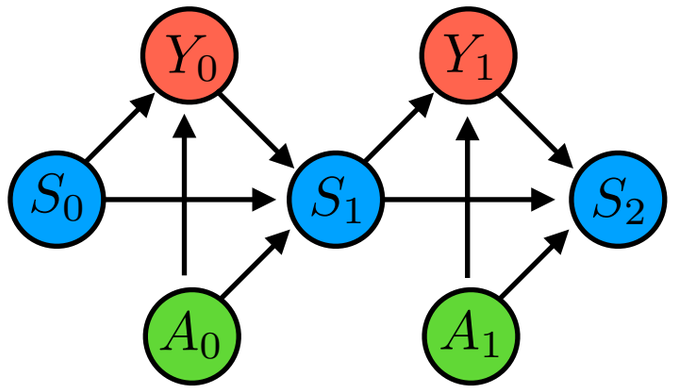


- Trivial transducers $S_t = 0$ <—> Memoryless interfaces $p(\boldsymbol{y}_{:t}|\boldsymbol{a}_{:}) = \prod_{\tau=0} p(y_\tau|a_\tau)$
- Fully observability $S_t = Y_t$ <—> Markov interfaces $p(y_{t+1}|\boldsymbol{y}_{:t}, \boldsymbol{a}_{:}) = p(y_{t+1}|y_t, a_t)$

Transducers capture the memory of interfaces!

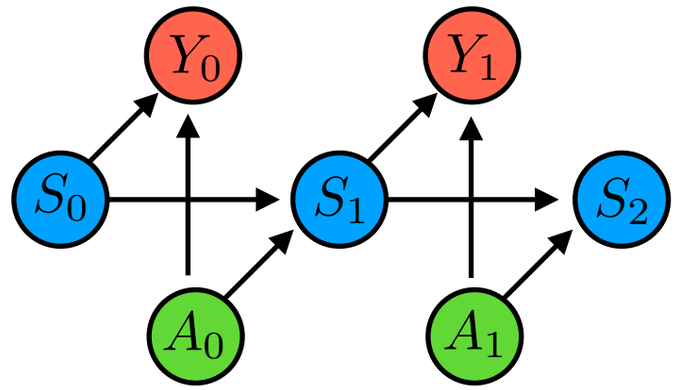
Different types of transducers

Full transducer



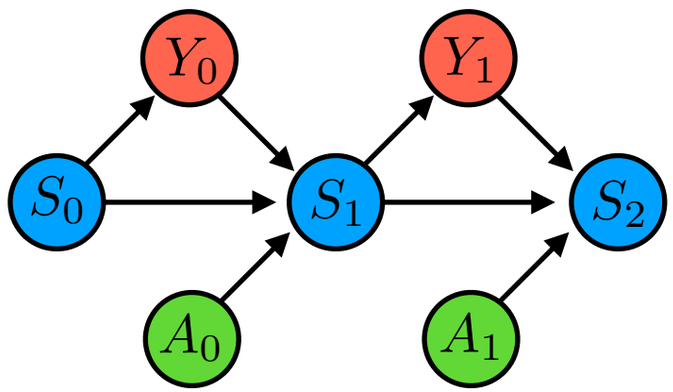
$\kappa_\tau(y, \tilde{s}|a, s)$

POMDP — action first



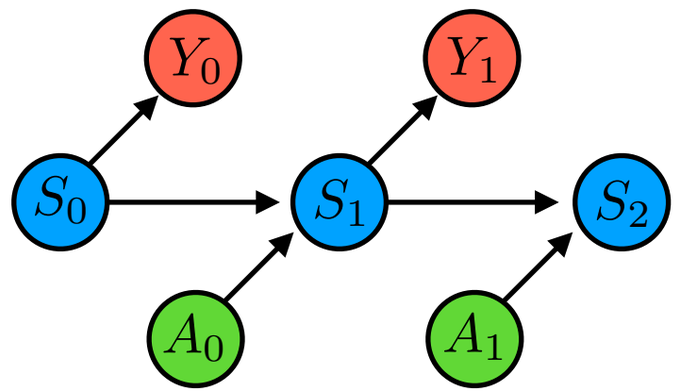
$\kappa_\tau(y, \tilde{s}|a, s) = \mu_\tau(y|a, s)\nu_\tau(\tilde{s}|a, s)$

Belief transducer



$\kappa_\tau(y, \tilde{s}|a, s) = \mu_\tau(y|s)\nu_\tau(\tilde{s}|y, a, s)$

POMDP — observation first



$\kappa_\tau(y, \tilde{s}|a, s) = \mu_\tau(y|s)\nu_\tau(\tilde{s}|a, s)$

Contents

1. Hierarchical emergence

2. World modelling

a) Setting and formal foundations

b) Minimal world models

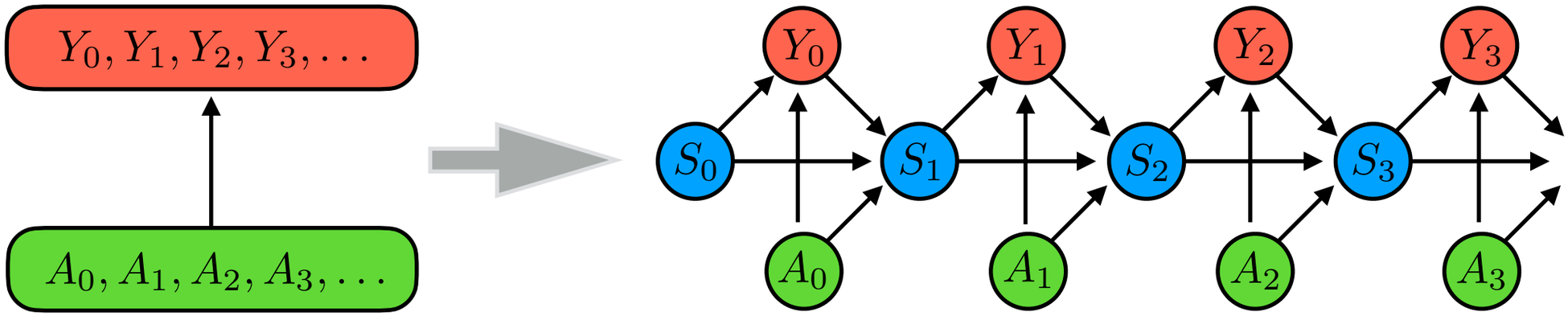
c) Interpretable world models

3. Abstractions

4. Ideas to take home

Reducing transducers

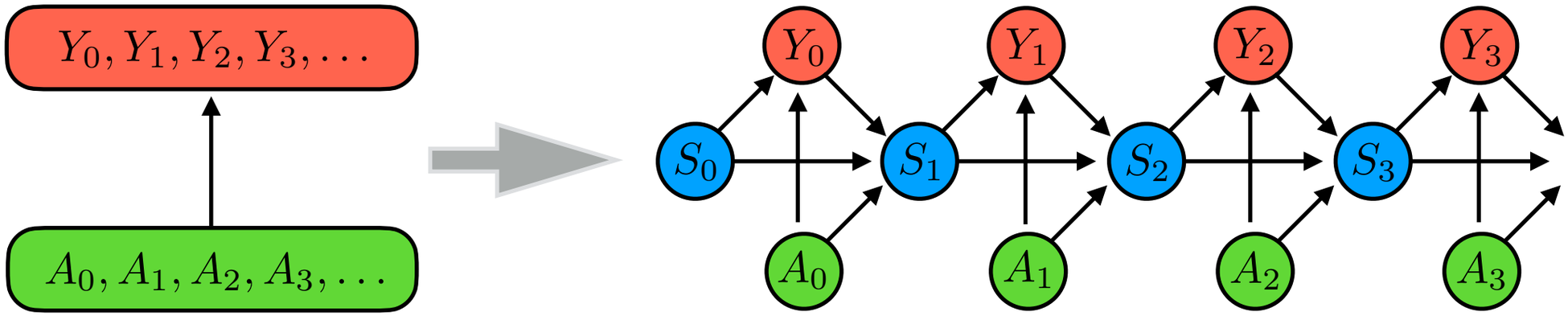
Can all interfaces be simulated via a transducer?



$$p(\boldsymbol{y}_{:t}\boldsymbol{s}_{:t+1}|\boldsymbol{a}_{:}) = p(s_0)\prod_{\tau=0}^{t}\kappa_\tau(y_\tau, s_{\tau+1}|a_\tau, s_t)$$

Reducing transducers

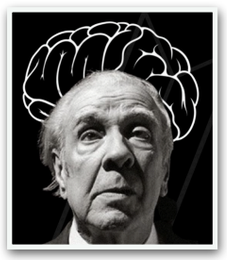
Can all interfaces be simulated via a transducer?



$$p(\boldsymbol{y}_{:t}\boldsymbol{s}_{:t+1}|\boldsymbol{a}_{:}) = p(s_0)\prod_{\tau=0}^{t}\kappa_\tau(y_\tau, s_{\tau+1}|a_\tau, s_t)$$

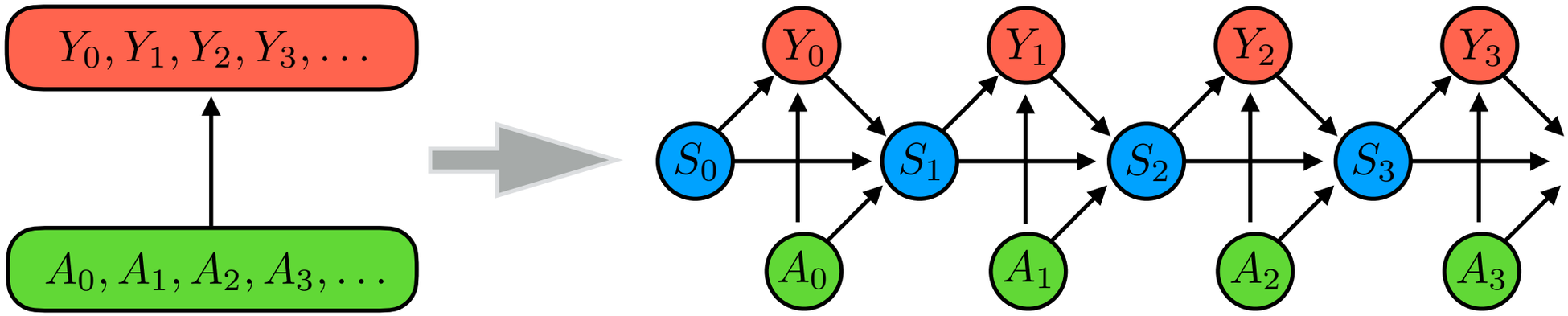
Yes! Consider the transducer "Funes" with world model $S_t = \boldsymbol{H}_{:t-1}$.

$$S_0 = 0, \quad S_1 = (A_0, Y_0), \quad S_2 = (\boldsymbol{A}_{:1}, \boldsymbol{Y}_{:1}), \ldots$$



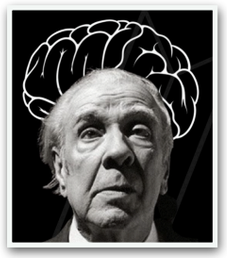
Reducing transducers

Can all interfaces be simulated via a transducer?



$$p(\boldsymbol{y}_{:t}\boldsymbol{s}_{:t+1}|\boldsymbol{a}_{:}) = p(s_0)\prod_{\tau=0}^{t}\kappa_\tau(y_\tau, s_{\tau+1}|a_\tau, s_t)$$

Yes! Consider the transducer "Funes" with world model $S_t = \boldsymbol{H}_{:t-1}$.

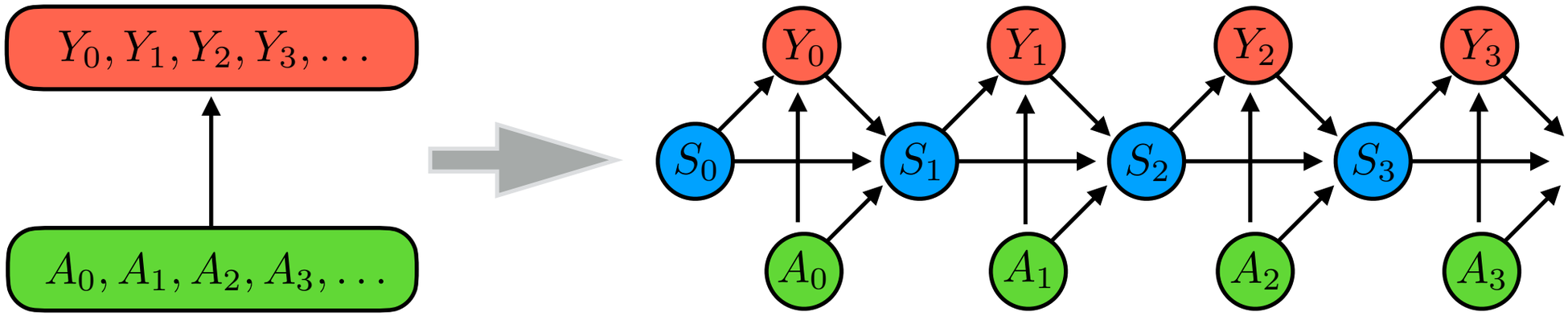


$$S_0 = 0, \quad S_1 = (A_0, Y_0), \quad S_2 = (\boldsymbol{A}_{:1}, \boldsymbol{Y}_{:1}), \ldots$$

—> It works, but requires exponential amounts of memory

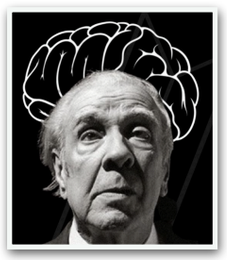
Reducing transducers

Can all interfaces be simulated via a transducer?



$$p(\boldsymbol{y}_{:t}\boldsymbol{s}_{:t+1}|\boldsymbol{a}_{:}) = p(s_0)\prod_{\tau=0}^{t}\kappa_\tau(y_\tau, s_{\tau+1}|a_\tau, s_t)$$

Yes! Consider the transducer "Funes" with world model $S_t = \boldsymbol{H}_{:t-1}$.



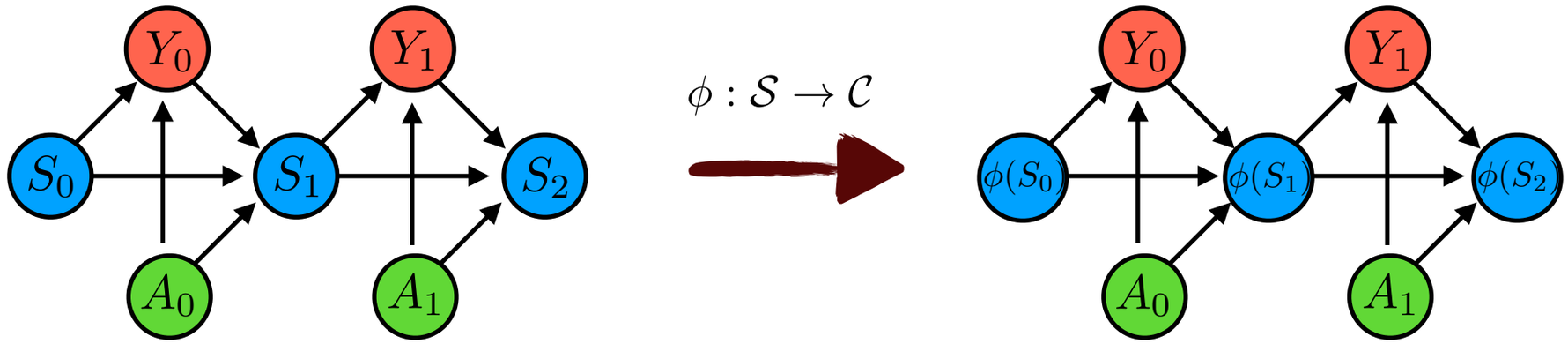
$$S_0 = 0, \quad S_1 = (A_0, Y_0), \quad S_2 = (\boldsymbol{A}_{:1}, \boldsymbol{Y}_{:1}), \ldots$$

—> It works, but requires exponential amounts of memory

How can one "**reduce**" a transducer?

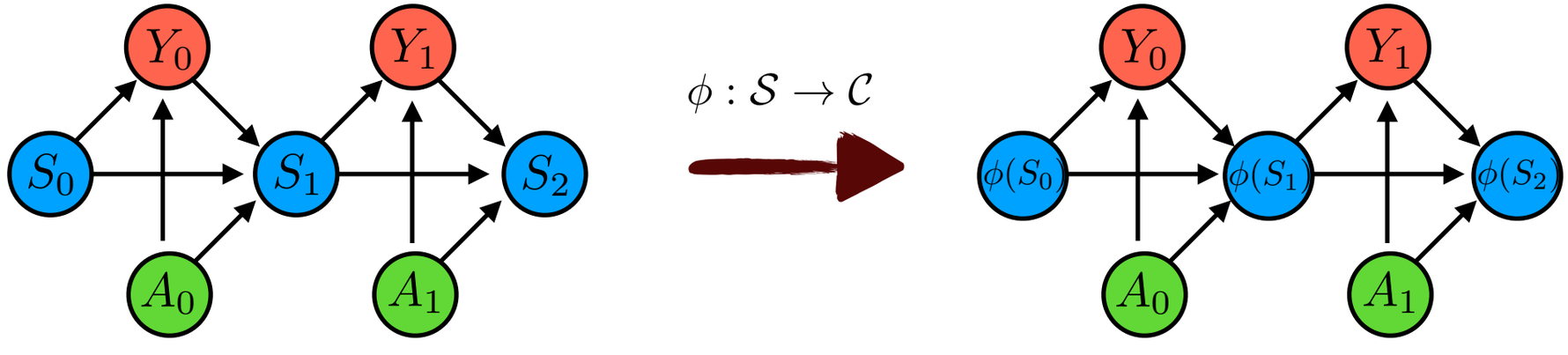
Reducing transducers: tools

World reductions: coarse-grains that keep the interface stays the same (equivalent to **bisimulation**).

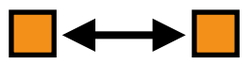


Reducing transducers: tools

World reductions: coarse-grains that keep the interface stays the same (equivalent to **bisimulation**).

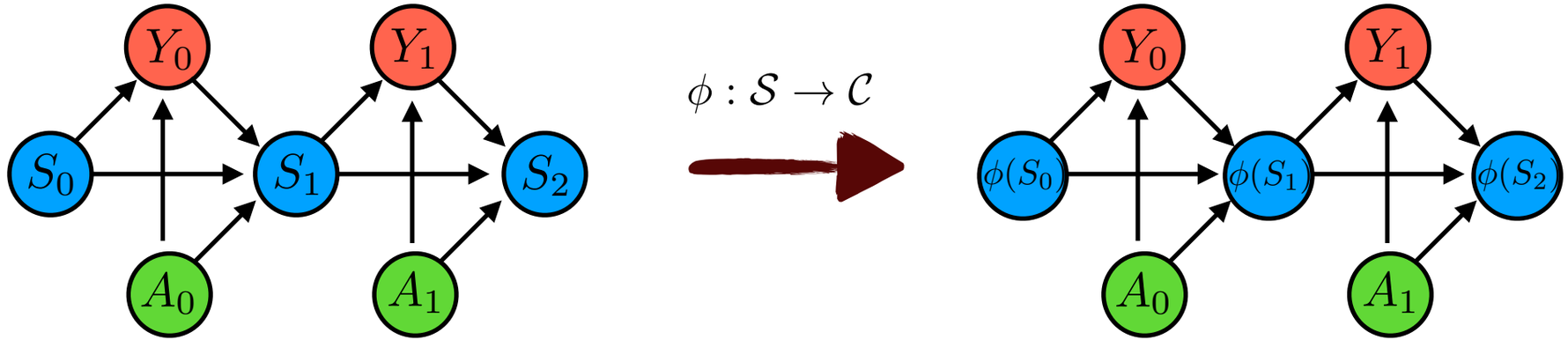


Isomorphic worlds: it is possible to reduce one into each other.



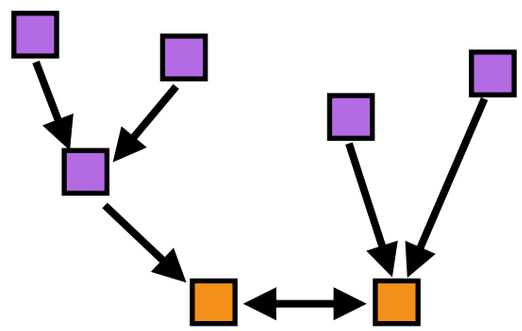
Reducing transducers: tools

World reductions: coarse-grains that keep the interface stays the same (equivalent to **bisimulation**).



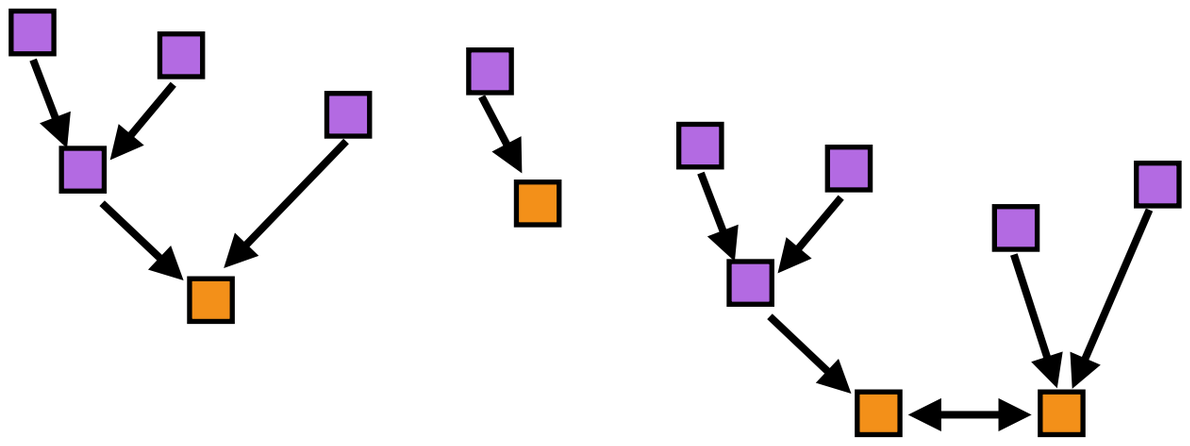
Isomorphic worlds: it is possible to reduce one into each other.

Minimal worlds: all reductions are isomorphic to itself.



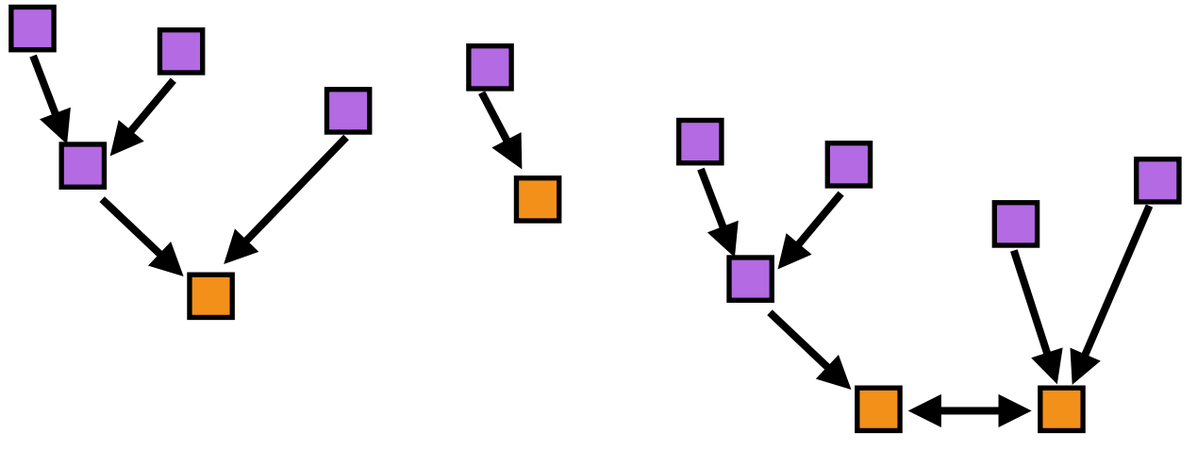
Limitations of bisimulation

Problem: bisimulation usually doesn't have a global minima



Limitations of bisimulation

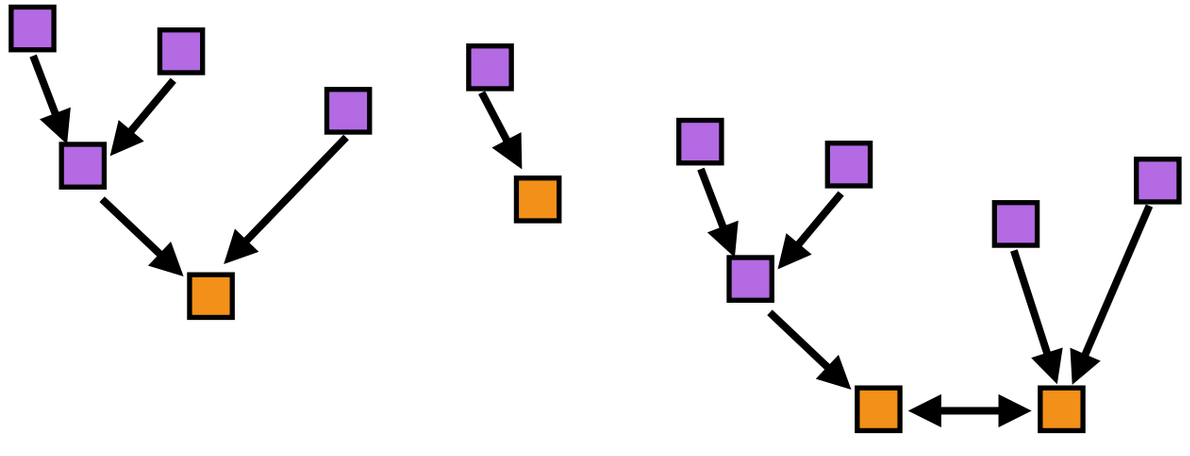
Problem: bisimulation usually doesn't have a global minima



Reason: bisimulation can replace pairwisely identical states, but cannot compress redundancy between groups of three or more states…

Limitations of bisimulation

Problem: bisimulation usually doesn't have a global minima



Reason: bisimulation can replace pairwisely identical states, but cannot compress redundancy between groups of three or more states…

Solution: accept what you get, or embrace negative probabilities!

Generalised transducers

A quasi-stochastic vector sums 1 but can have negative numbers.

$$p = (0.6, -0.1, 0.5)$$

Generalised transducers

A quasi-stochastic vector sums 1 but can have negative numbers.

Generalised transducers have quasi-stochastic states and transitions:

$$\Pr(\boldsymbol{y}_{:t}|\boldsymbol{a}_{:t}) = \boldsymbol{u}^\intercal \cdot \left(\prod_{i=0}^{t} A_i^{(y_i|a_i)} \right) \cdot \boldsymbol{v}$$

Generalised transducers

A quasi-stochastic vector sums 1 but can have negative numbers.

Generalised transducers have quasi-stochastic states and transitions:

$$\Pr(\boldsymbol{y}_{:t}|\boldsymbol{a}_{:t}) = \boldsymbol{u}^\intercal \cdot \left(\prod_{i=0}^{t} A_i^{(y_i|a_i)} \right) \cdot \boldsymbol{v}$$

Theorem: generalised transducers can always be reduced up to a unique minimum!

Generalised transducers

A quasi-stochastic vector sums 1 but can have negative numbers.

Generalised transducers have quasi-stochastic states and transitions:

$$\Pr(\boldsymbol{y}_{:t}|\boldsymbol{a}_{:t}) = \boldsymbol{u}^\intercal \cdot \left(\prod_{i=0}^{t} A_i^{(y_i|a_i)} \right) \cdot \boldsymbol{v}$$

Theorem: generalised transducers can always be reduced up to a unique minimum!

Limitation: substantial lack of interpretability, inability to sample, etc

Contents
1. Hierarchical emergence
2. World modelling
a) Setting and formal foundations
b) Minimal world models
c) Interpretable world models
3. Abstractions
4. Ideas to take home

Interpretable models: what can agents learn?

What makes a world model learnable?

1. *Observable*: world's state is a (deterministic) function of the history, so $S_t = f(\boldsymbol{H}_{:t-1})$
2. *Unifilar*: world state can be efficiently updated as $S_{t+1} = \hat{f}(S_t, Y_t, A_t)$

Interpretable models: what can agents learn?

What makes a world model learnable?

1. *Observable*: world's state is a (deterministic) function of the history, so $S_t = f(\boldsymbol{H}_{:t-1})$
2. *Unifilar*: world state can be efficiently updated as $S_{t+1} = \hat{f}(S_t, Y_t, A_t)$

How to create observable world models?

—-> By switching the phase space from states to distributions ("beliefs")

Interpretable models: what can agents learn?

What makes a world model learnable?

1. *Observable*: world's state is a (deterministic) function of the history, so $S_t = f(\boldsymbol{H}_{:t-1})$
2. *Unifilar*: world state can be efficiently updated as $S_{t+1} = \hat{f}(S_t, Y_t, A_t)$

How to create observable world models?

—-> By switching the phase space from states to distributions ("beliefs")

Bayesian beliefs are unifilar and yield a transducer that generate the same interface!

$$b_{t+1}(s_{t+1}) = \frac{1}{Z} \sum_{s_t} p(y_t, s_{t+1} | a_t, s_t) b_t(s_t)$$

Minimal predictive models

Let's build equivalence classes of histories where $\boldsymbol{h}_{:t-1} \sim_\epsilon \boldsymbol{h}'_{:t-1}$ if and only if

$$p(\boldsymbol{y}_{t:t+T}|\boldsymbol{h}_{:t-1}, \boldsymbol{a}_{t:t+T}) = p(\boldsymbol{y}_{t:t+T}|\boldsymbol{h}'_{:t-1}, \boldsymbol{a}_{t:t+T}), \quad \forall \boldsymbol{y}_{t:t+L}, \boldsymbol{a}_{t:t+L}, L \in \mathbb{N}.$$

These are the states of the "e-transducer".

Minimal predictive models

Let's build equivalence classes of histories where $\boldsymbol{h}_{:t-1} \sim_\epsilon \boldsymbol{h}'_{:t-1}$ if and only if

$$p(\boldsymbol{y}_{t:t+T}|\boldsymbol{h}_{:t-1}, \boldsymbol{a}_{t:t+T}) = p(\boldsymbol{y}_{t:t+T}|\boldsymbol{h}'_{:t-1}, \boldsymbol{a}_{t:t+T}), \quad \forall \boldsymbol{y}_{t:t+L}, \boldsymbol{a}_{t:t+L}, L \in \mathbb{N}.$$

These are the states of the "e-transducer".

Findings:

1. It is the minimal bisimulation of the Funes transducer
2. It is a transducer.
3. It is the minimal bisimulation of ***any*** predictive world model that generates the interface.
4. Thus, it is the minimal predictive world model that generates the interface.

Minimal predictive models

Let's build equivalence classes of histories where $\boldsymbol{h}_{:t-1} \sim_\epsilon \boldsymbol{h}'_{:t-1}$ if and only if

$$p(\boldsymbol{y}_{t:t+T}|\boldsymbol{h}_{:t-1}, \boldsymbol{a}_{t:t+T}) = p(\boldsymbol{y}_{t:t+T}|\boldsymbol{h}'_{:t-1}, \boldsymbol{a}_{t:t+T}), \quad \forall \boldsymbol{y}_{t:t+L}, \boldsymbol{a}_{t:t+L}, L \in \mathbb{N}.$$

These are the states of the "e-transducer".

Findings:

1. It is the minimal bisimulation of the Funes transducer
2. It is a transducer.
3. It is the minimal bisimulation of ***any*** predictive world model that generates the interface.
4. Thus, it is the minimal predictive world model that generates the interface.

Conclusion 1: agents with same interface but beliefs in different world models still share the e-transducer.

Minimal predictive models

Let's build equivalence classes of histories where $\boldsymbol{h}_{:t-1} \sim_\epsilon \boldsymbol{h}'_{:t-1}$ if and only if

$$p(\boldsymbol{y}_{t:t+T}|\boldsymbol{h}_{:t-1}, \boldsymbol{a}_{t:t+T}) = p(\boldsymbol{y}_{t:t+T}|\boldsymbol{h}'_{:t-1}, \boldsymbol{a}_{t:t+T}), \quad \forall \boldsymbol{y}_{t:t+L}, \boldsymbol{a}_{t:t+L}, L \in \mathbb{N}.$$

These are the states of the "e-transducer".

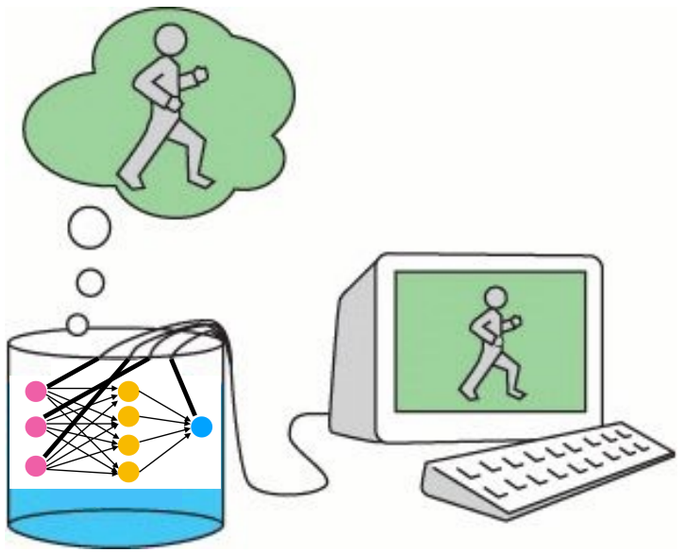
Findings:

1. It is the minimal bisimulation of the Funes transducer
2. It is a transducer.
3. It is the minimal bisimulation of ***any*** predictive world model that generates the interface.
4. Thus, it is the minimal predictive world model that generates the interface.

Conclusion 1: agents with same interface but beliefs in different world models still share the e-transducer.

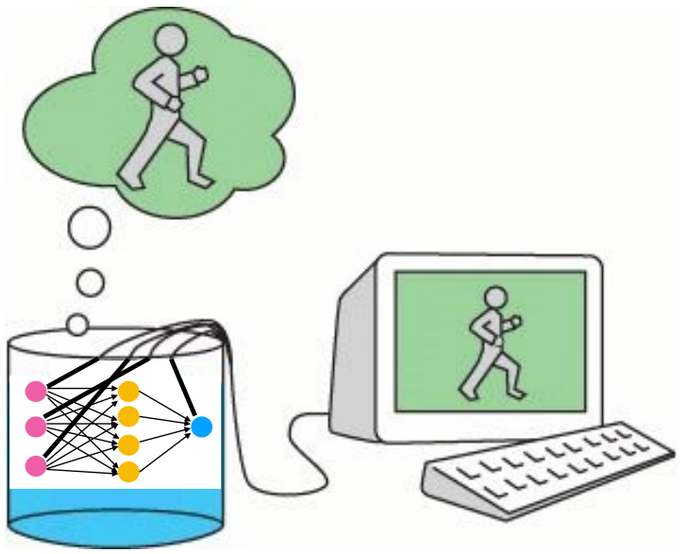
Conclusion 2: the e-transducer contains all the predictive information about the world that is learnable from the interface.

What have we learned?



- The class of world models that can be used to simulate a given agent's interface is *surprisingly rich*.

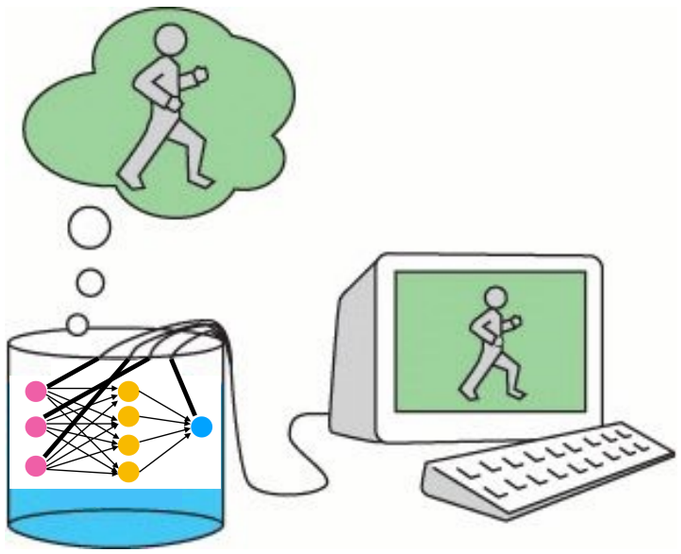
What have we learned?



- The class of world models that can be used to simulate a given agent's interface is *surprisingly rich*.

- There exists a fundamental trade-off between computational efficiency and interpretability of world models.

What have we learned?



- The class of world models that can be used to simulate a given agent's interface is *surprisingly rich*.

- There exists a fundamental trade-off between computational efficiency and interpretability of world models.

- Bayesian belief updating is the most efficient world model that an agent can compute for itself.

Contents

1. Hierarchical emergence

2. World modelling

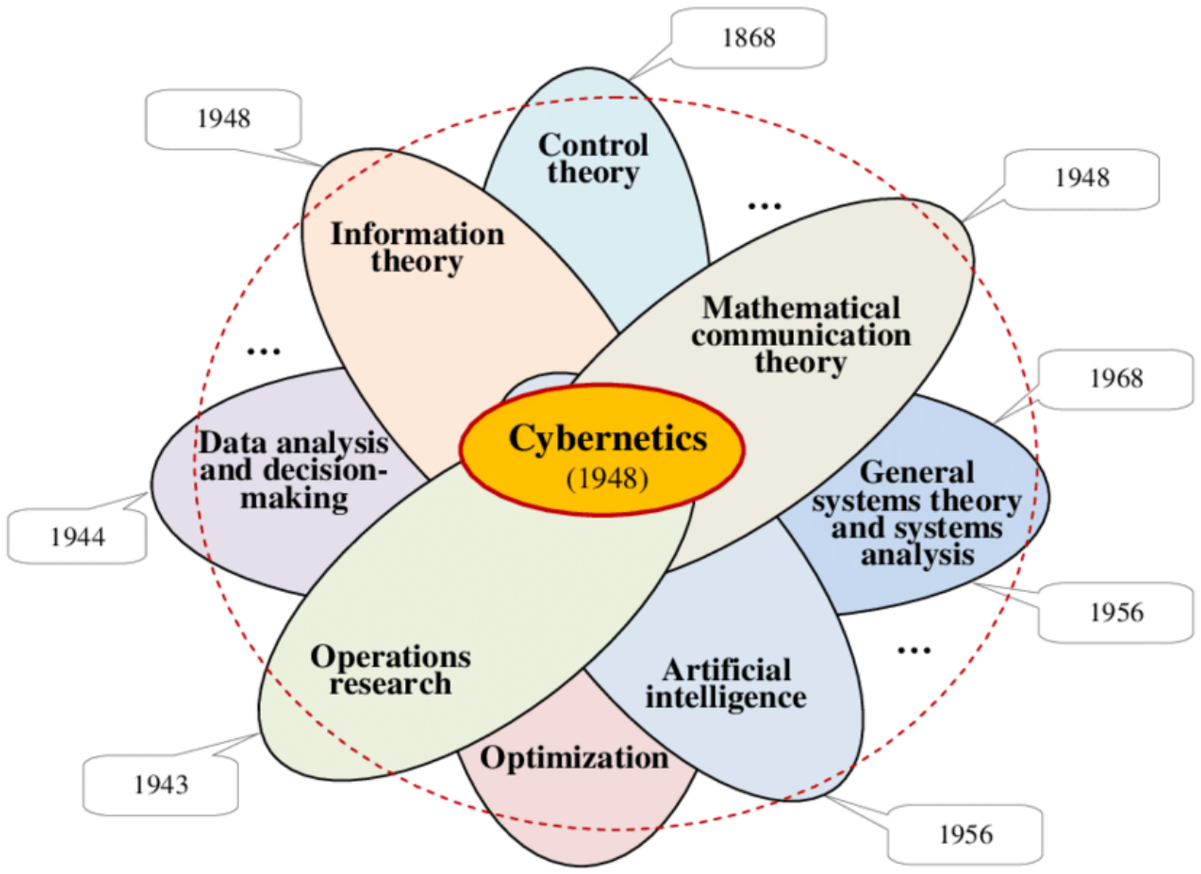
3. Abstractions

4. Ideas to take home

A cybernetic-inspired approach to world modelling

Cybernetics: very ambitious research programme to explain intelligent behaviour.

Lost momentum but planted seeds to many disciplines (control theory, AI, cognitive science…)



A cybernetic-inspired approach to world modelling

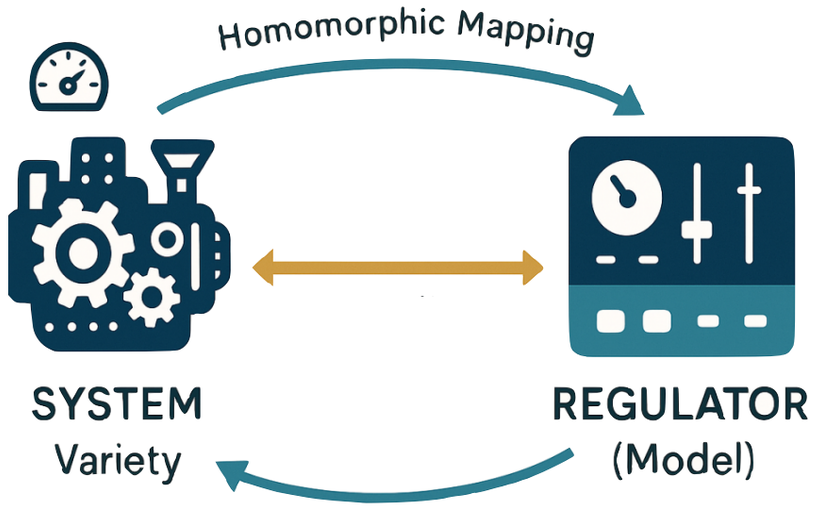
Cybernetics: very ambitious research programme to explain intelligent behaviour.

Lost momentum but planted seeds to many disciplines (control theory, AI, cognitive science…)

Some good ideas:

— **Principle of requisite variety:** a controller needs at least the same diversity as what its target

— **Good regulator theorem:** a good controller needs a model of its target



https://www.linkedin.com/pulse/clarifying-good-regulator-theorem-through-rosens-modeling-lissack-6aqtf/

Ashby, W. R. (1958). "Requisite variety and its implications for the control of complex systems." Cybernetica, 1(2), 83-99.

Conant & Ashby (1970). "Every good regulator of a system must be a model of that system." International Journal of Systems Science, 1(2), 89–97.

A cybernetic-inspired approach to world modelling

Cybernetics: very ambitious research programme to explain intelligent behaviour.

Lost momentum but planted seeds to many disciplines (control theory, AI, cognitive science…)

Some good ideas:

— **Principle of requisite variety:** a controller needs at least the same diversity as what its target

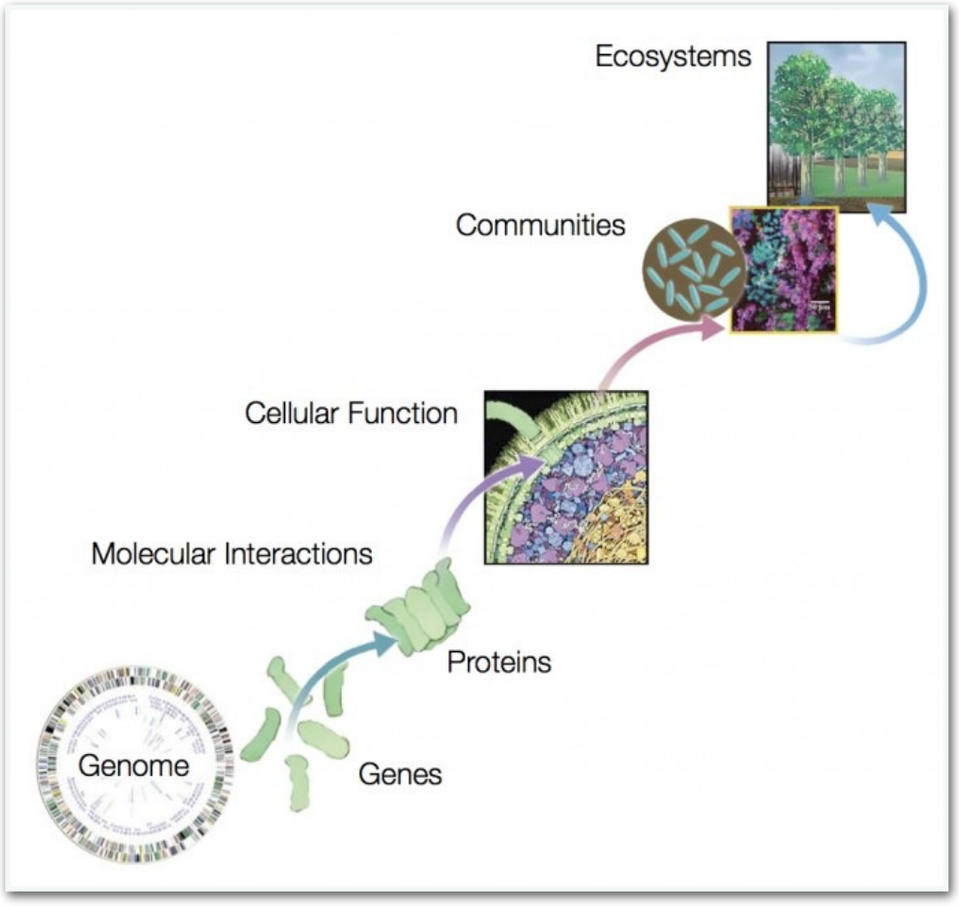
— **Good regulator theorem:** a good controller needs a model of its target

Interesting question:
To what degree an agents' beliefs need to reflect structural properties of their environments

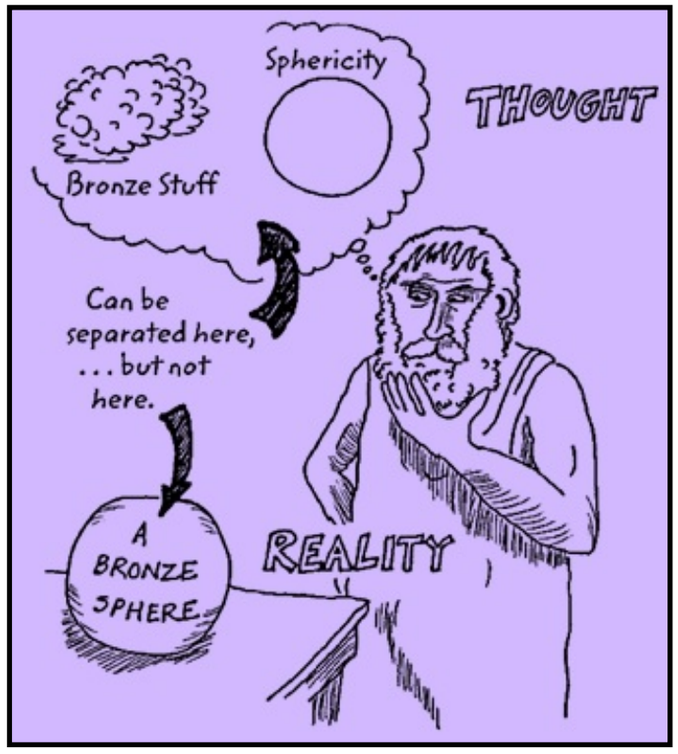
Key idea: “Emergence matching”

Exploring the conditions under which physical processes with **hierarchical structure** give rise to **hierarchical beliefs**.

Physical process

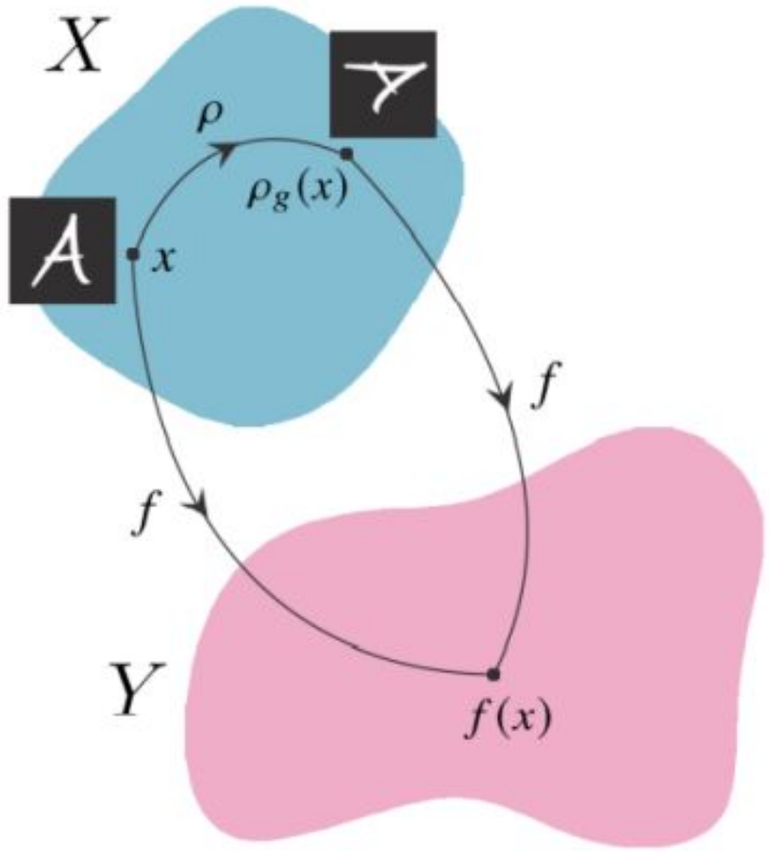


World models



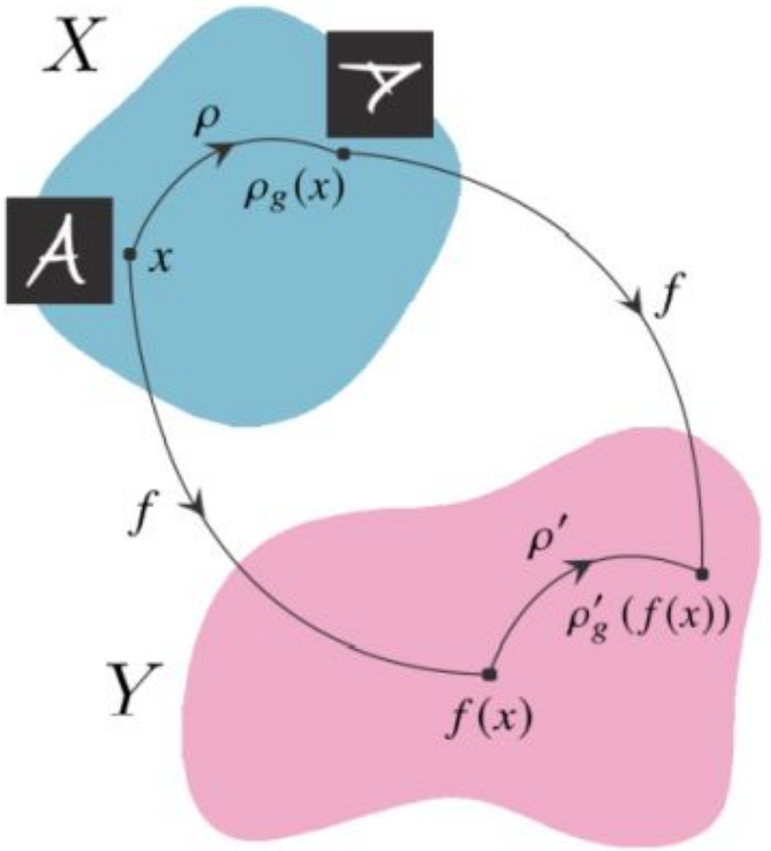
Rosas (2025). Symmetries at the origin of hierarchical emergence. arXiv preprint arXiv:2512.00984.

Symmetry and emergence?



Invariance

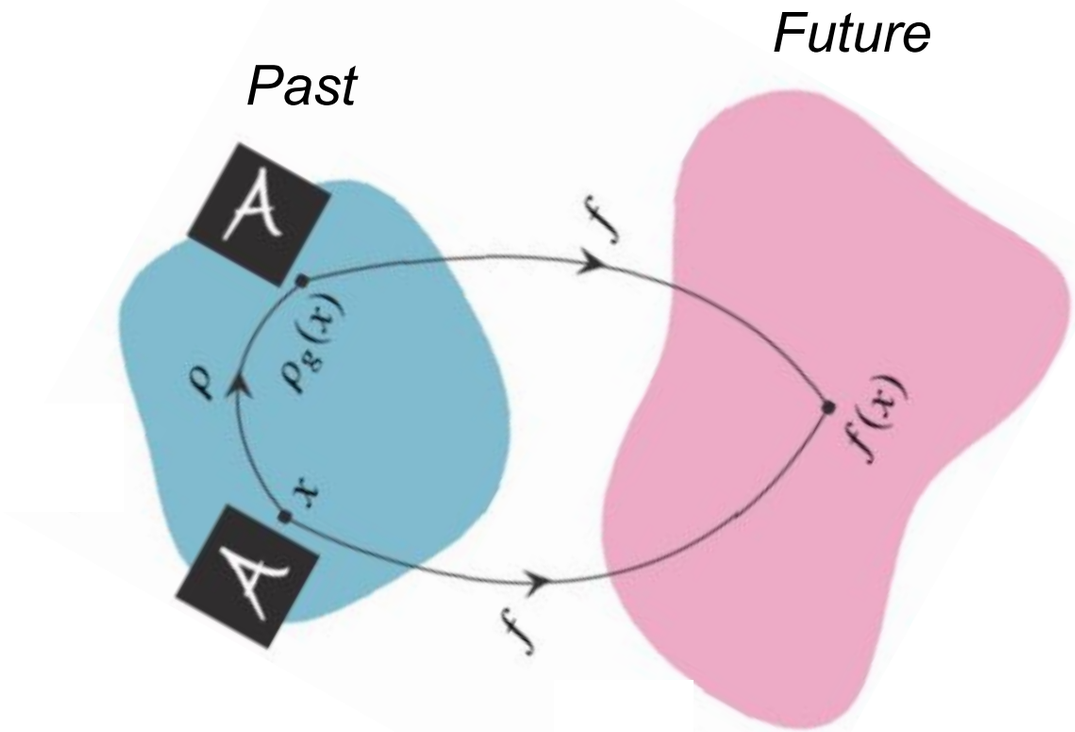
$$f(\rho_g(x)) = f(x)$$



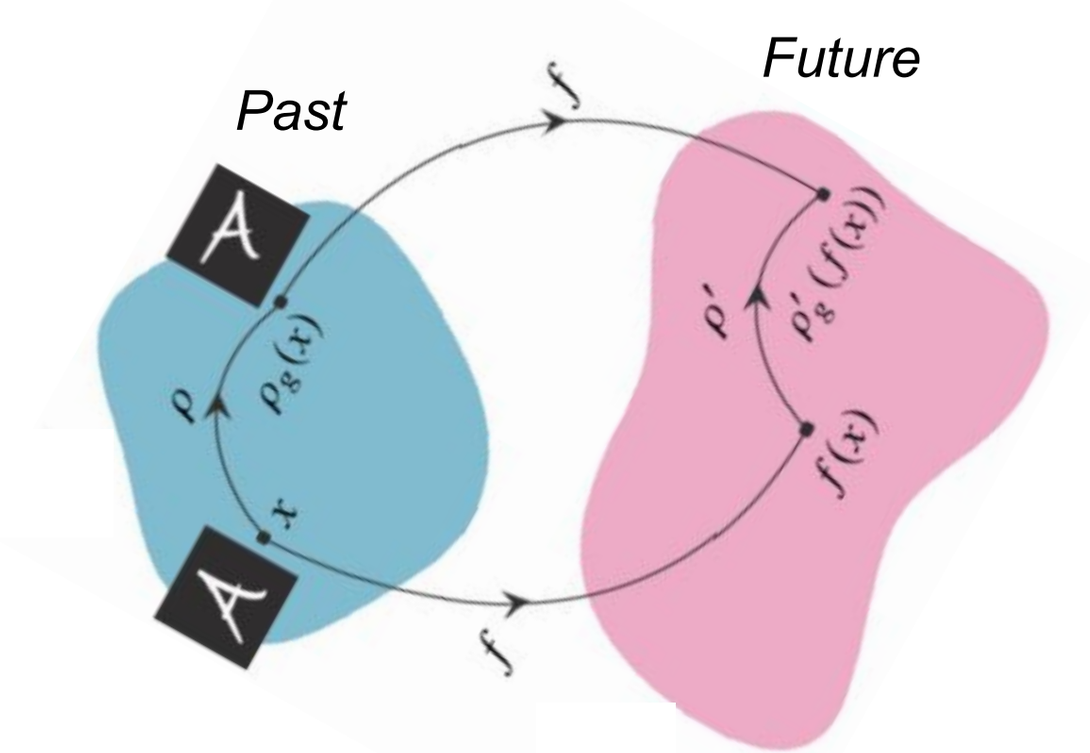
Equivariance

$$f(\rho_g(x)) = \rho'_g(f(x))$$

Symmetry and emergence?



Invariant dynamics
—> symmetric
initial conditions
lead to same future

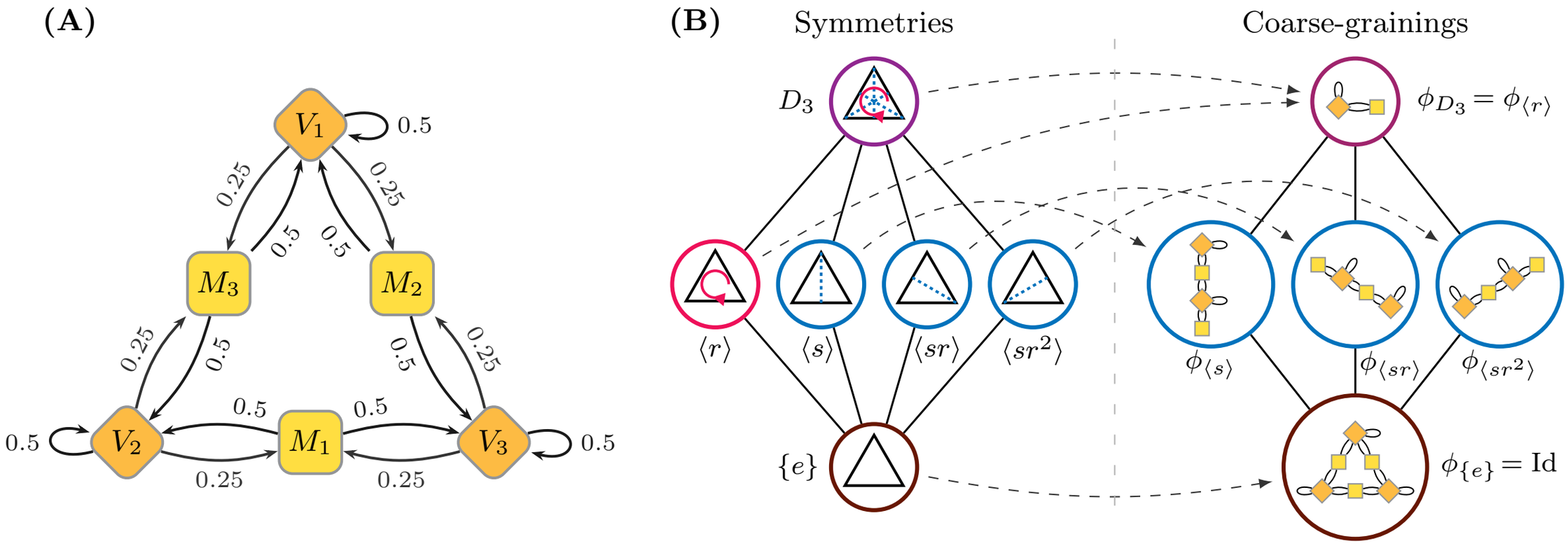


Equivariant dynamics
—> symmetric
initial conditions
lead to symmetric futures

Examples: particle exchangeability, left-right symmetry, etc

Symmetry as sufficient for hierarchical emergence

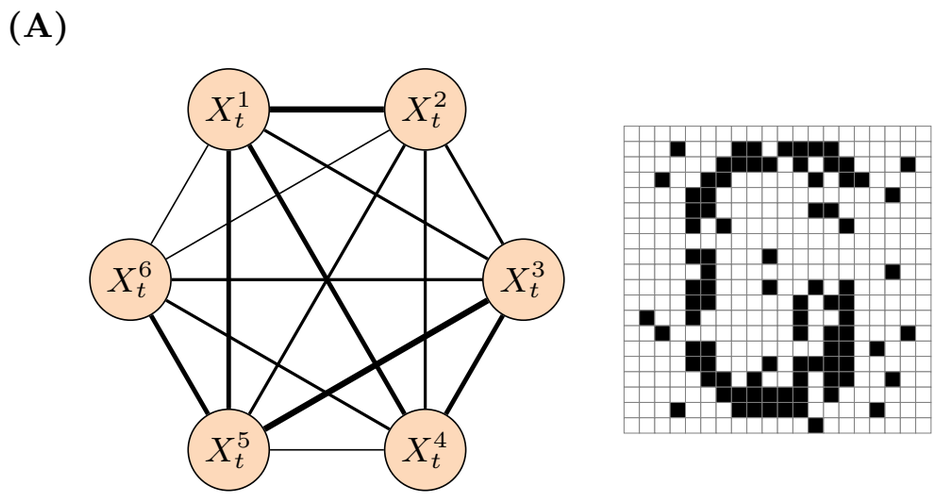
Theorem: if Markov process is dynamically equivariant
—> orbits are informationally closed



Rosas (2025). Symmetries at the origin of hierarchical emergence. arXiv preprint arXiv:2512.00984.

Example: Hopfield networks

Physical systems with equivariant dynamics have macroscopic levels that are informationally-closed.



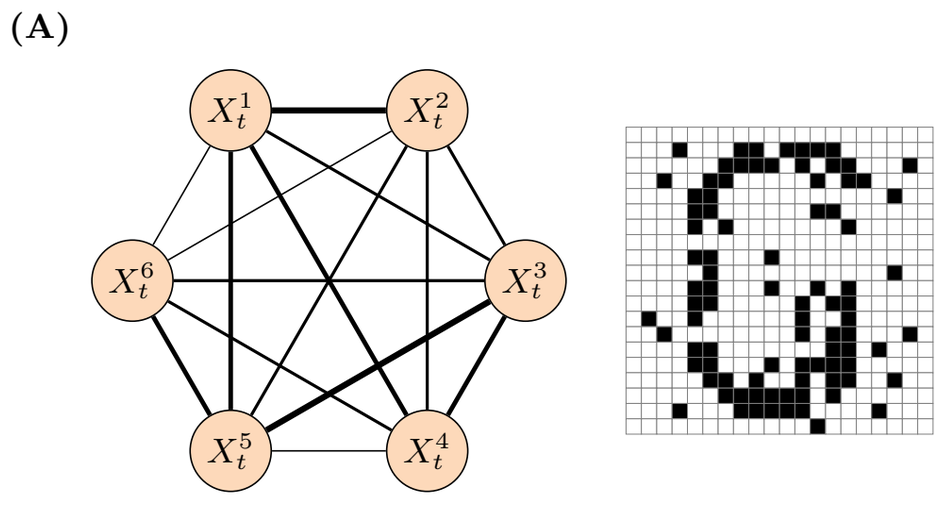
Neural dynamics: $\mathbb{P}\{X_{t+1}^i = 1 | X_t^1, \ldots, X_t^n\} = F\left(\beta \sum_{j=1}^{n} w_{i,j} X_t^j\right)$

Invariances: Mattis magnetisation $Z_t^\mu = \frac{1}{n}(q_\mu \cdot X_t) = \frac{1}{n} \sum_{j=1}^{n} q_\mu^j X_t^j$

—> Mattis Magnetisation is informationally closed!

Example: Hopfield networks

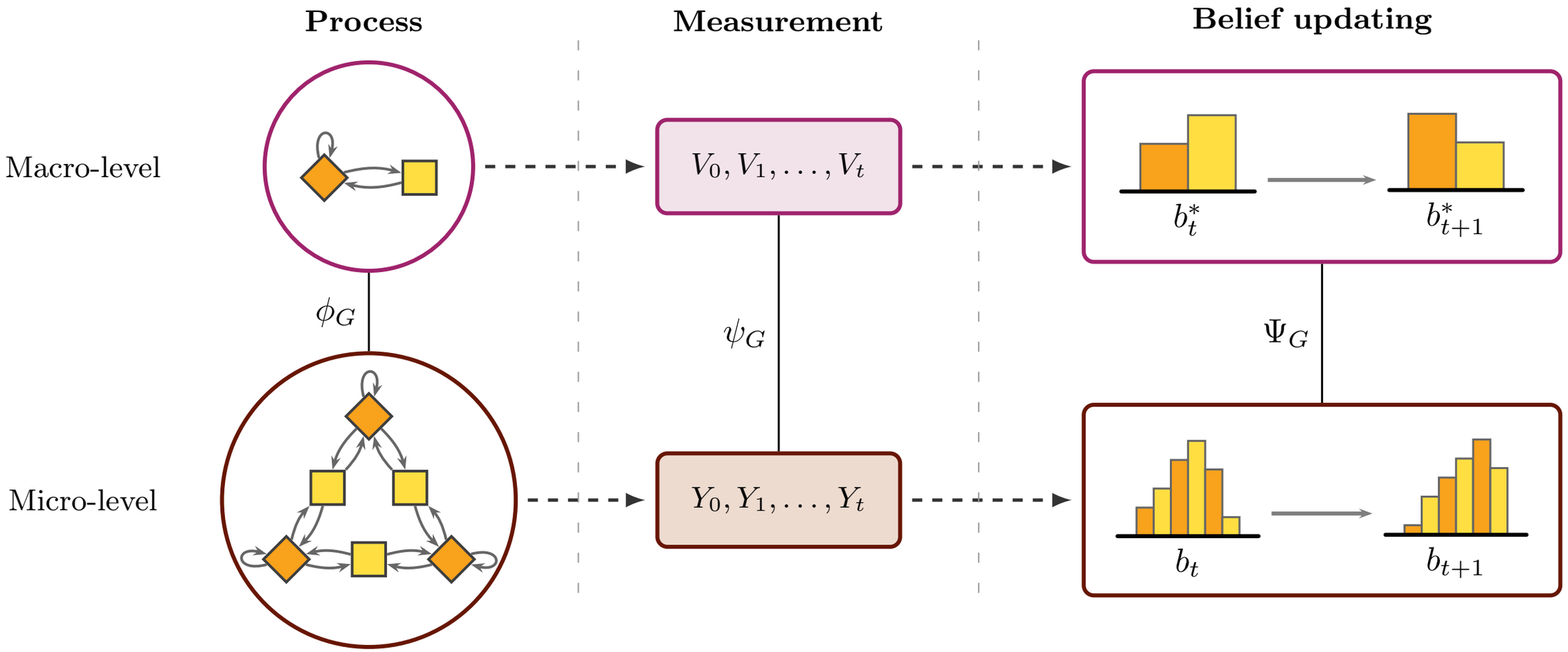
Physical systems with equivariant dynamics have macroscopic levels that are informationally-closed.



Compositionality is reflected in partial order!

Multi-level Bayesian belief updating

Consider a process with equivariant *dynamics* and also *measurements*.

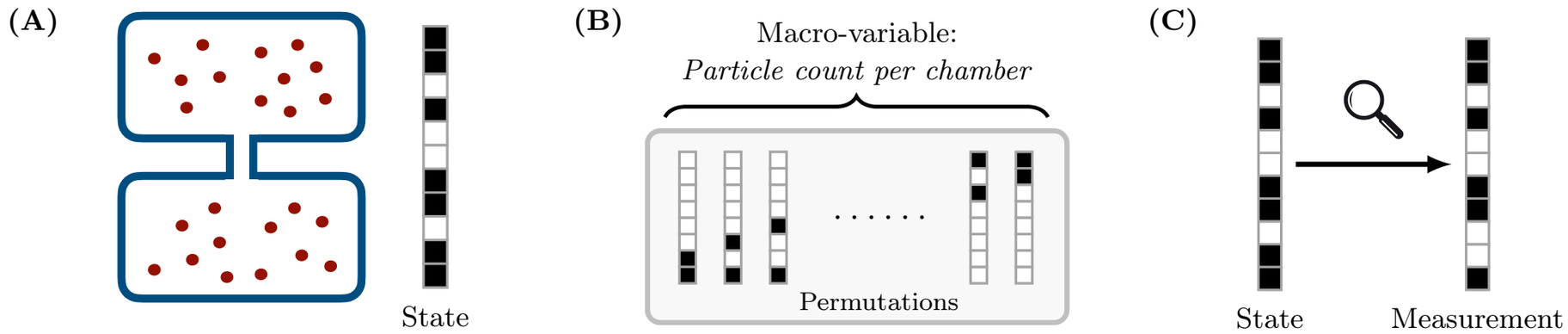


Theorem: Symmetry in (i) transitions and (ii) observations guarantees that Bayesian beliefs can be updated at multiple levels of resolution.

Coarse-grained beliefs —> Abstractions!

Example: Multi-level belief updating in Ehrenfest diffusion

Exact Bayesian updating on large systems is possible!!

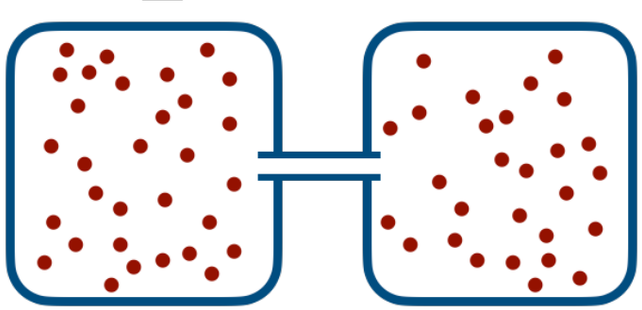


Current work: Thermodynamic reinforcement learning

The Ehrenfest control problem:

- An agent can push particles between chambers.
- Is rewarded for moving particles to a “good” chamber.
- Can choose to perceive and control at micro or macro levels.

$$Z_t = \sum X_t^k \qquad\qquad n - Z_t$$



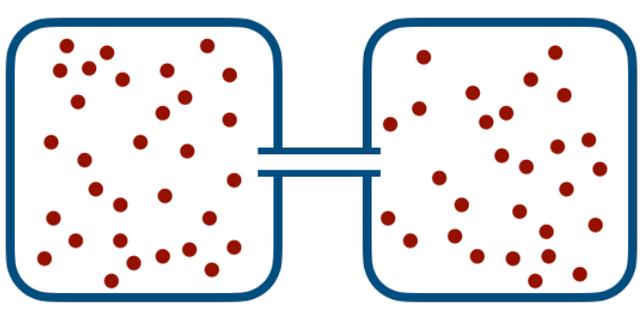
$$X_t = (0, 0, 1, 0, 1, \ldots, 1)$$

Current work: Thermodynamic reinforcement learning

The Ehrenfest control problem:

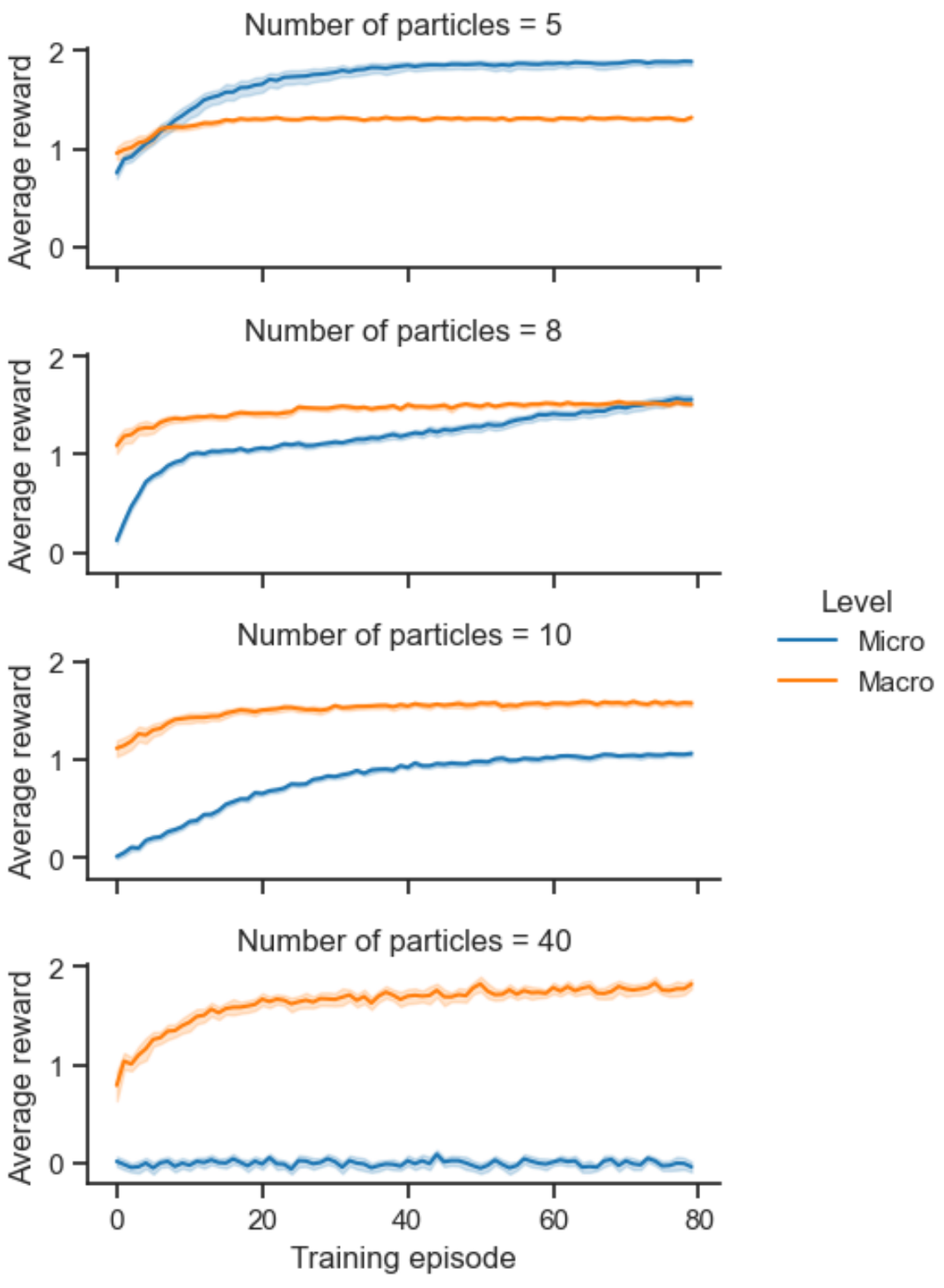
- An agent can push particles between chambers.
- Is rewarded for moving particles to a “good” chamber.
- Can choose to perceive and control at micro or macro levels.

$$Z_t = \sum X_t^k \qquad\qquad n - Z_t$$



$$X_t = (0, 0, 1, 0, 1, \ldots, 1)$$

For large systems micro-level control is unfeasible!



Contents

1. Hierarchical emergence

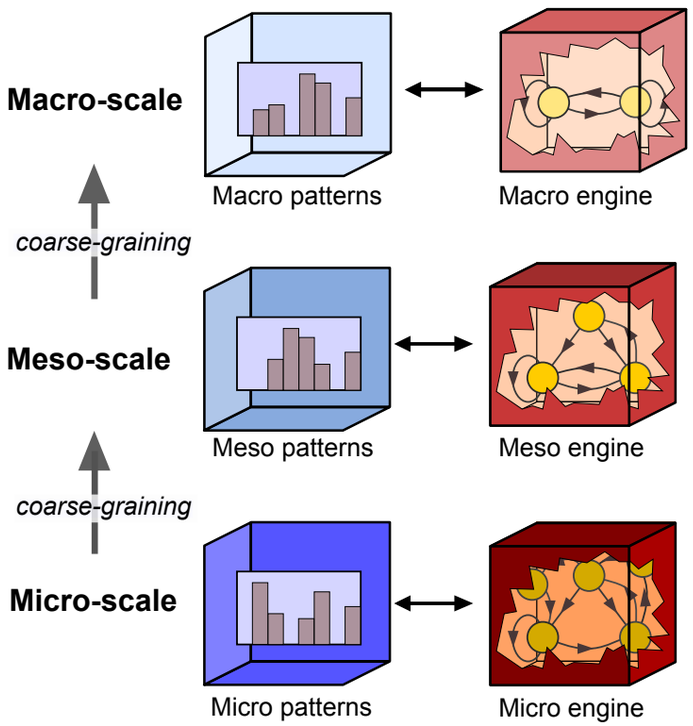
2. World modelling

3. Abstractions

4. Ideas to take home

Ideas to take home

- **Hierarchical emergence** takes place when there are levels of a system that are informationally or computationally autonomous.

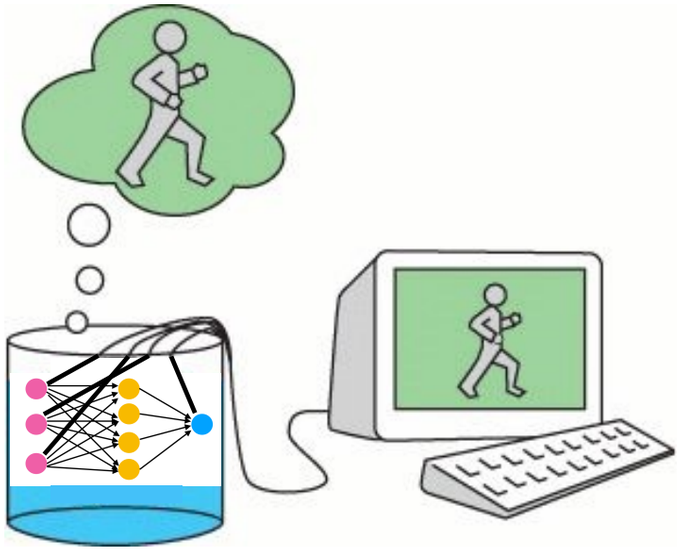


Why hierarchical emergence is cool?

1. ***For the system***: Enables decoupled levels with their own logic, which are amenable to efficient controllability.
2. ***For the researcher***: Allow for effective theories and efficient simulations.

Ideas to take home

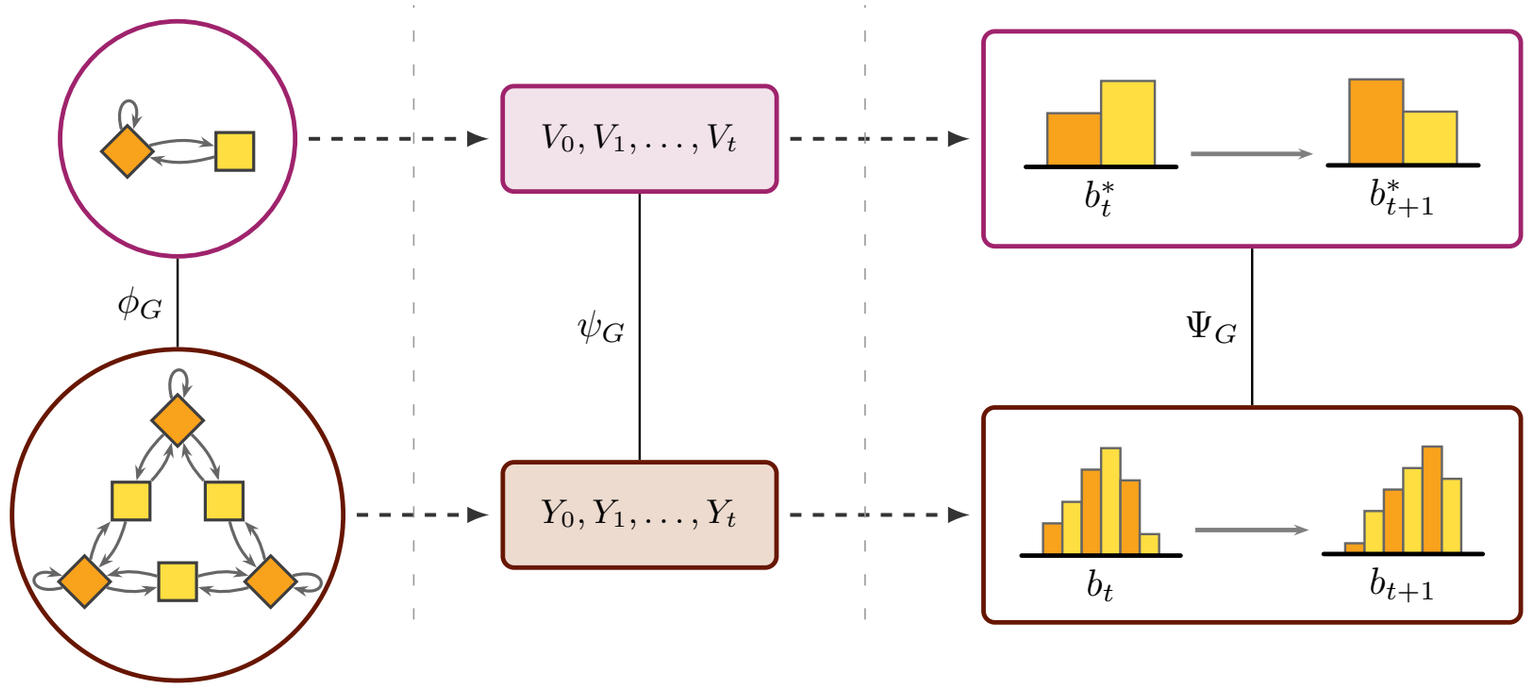
- **World modelling** corresponds to auxiliary processes that explain interfaces (i.e. perception-action loops).



- A given interface accepts many alternative world models.
- There is an intrinsic efficiency-vs- interpretability tradeoff
- Bayesian belief updating is the most efficient world model accessible for an agent.

Ideas to take home

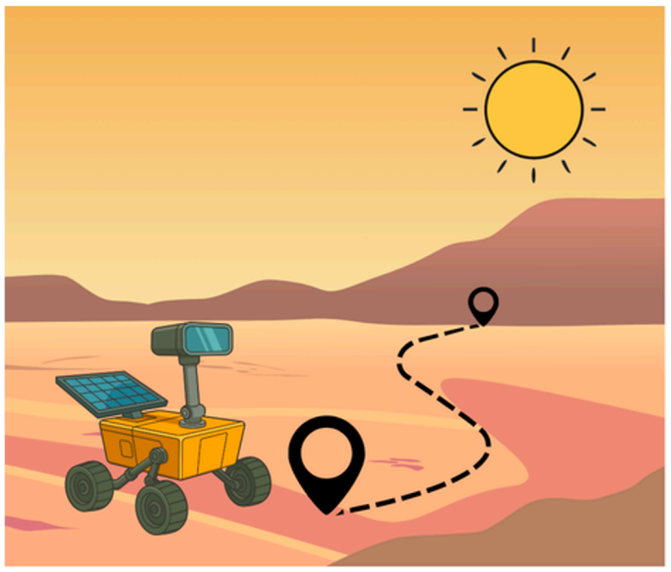
- **Dynamical equivariance** establishes a correspondence between hierarchical emergence and abstractions within belief updating.

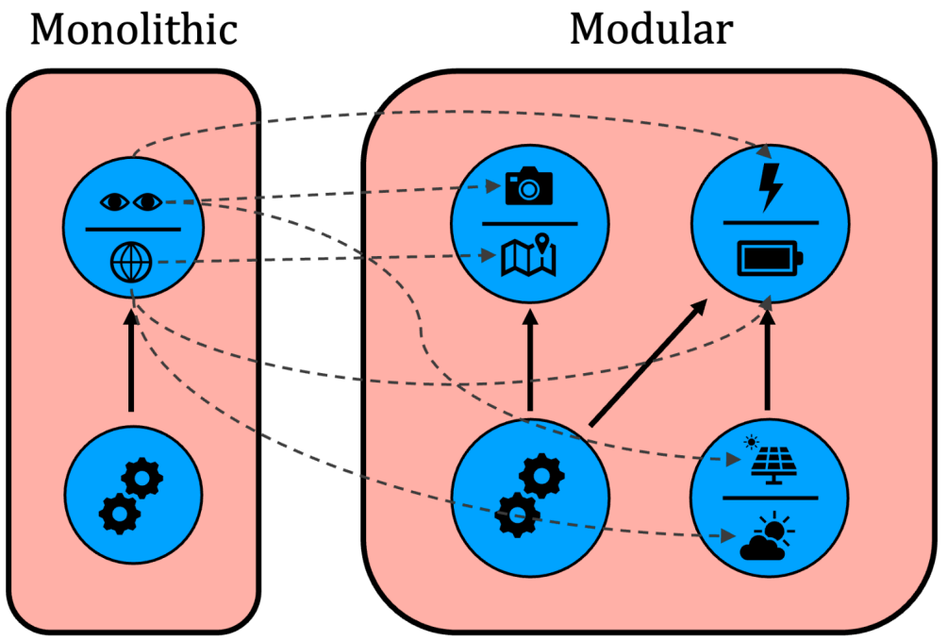


- Allows exact Bayesian inference in large systems.
- Lead to scalable approaches for RL in partially observable scenarios…

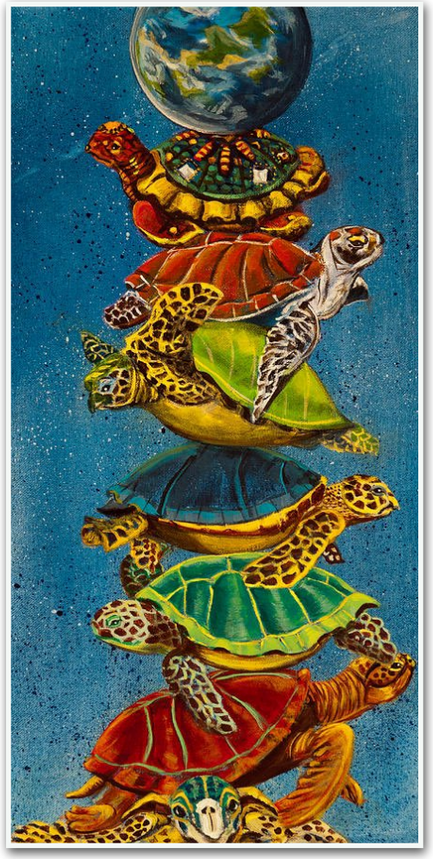
Ideas to take home

- Our current work is about combining these ideas with **compositionality**, and using them in **reinforcement learning** settings.





Boyd et al. (2025). “From monoliths to modules: Decomposing transducers for efficient world modelling”. arXiv preprint arXiv:2512.02193.



Thank you!

Contact information: Fernando E. Rosas

`f.rosas@sussex.ac.uk`