Building Minimal and Reusable Causal State Abstractions for Reinforcement Learning

(AAAI 2024)

Zizhao Wang*, **Caroline Wang***, Xuesu Xiao, Yuke Zhu, and Peter Stone





LARG Learning Agents Research Group The University of Texas at Austin | RPL | Sony AI

Problem Setup

Reinforcement Learning (RL) faces ongoing challenges, particularly in large state spaces

- sample inefficiency
- poor generalization

Solution: causal state abstractions

example task: grasp the pink bottle

Problem Setup

multiple tasks in the same environment
as *K* Markov decision processes:

$$(S, A, P,\ R^{1,\ldots,K})$$

non-redundant state space
$S = S^1 \times \cdots \times S^{d_S}$

action
space

unknown
transition

reward functions
for tasks $1, \ldots, K$

Problem Setup

State abstractions should be…

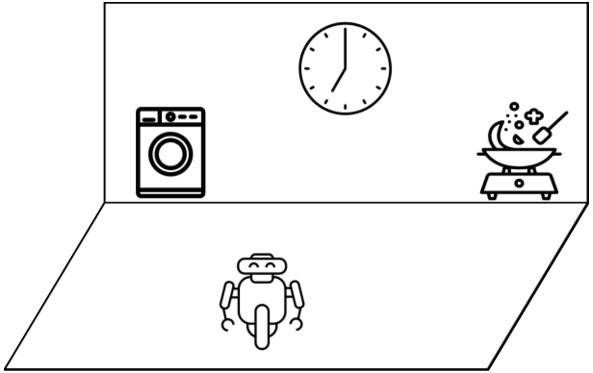
minimal and sufficient

- smallest input space for RL to learn a task
- improve sample efficiency & generalization

reusable

- enable the agent to learn all tasks in the same environment
- avoid learning each task from scratch

But how?



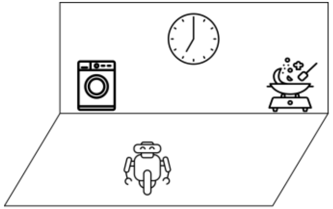
task 1: wash clothes in

$R_t^1 := \mathbb{1}[$  finishes the cycle $]$

task 2: cook dinner

$R_t^2 := \mathbb{1}[$ finish  at  $]$

Prior Work 1



CDL

Wang et al, "Causal dynamics learning for task-independent state abstraction" ICML 2022.

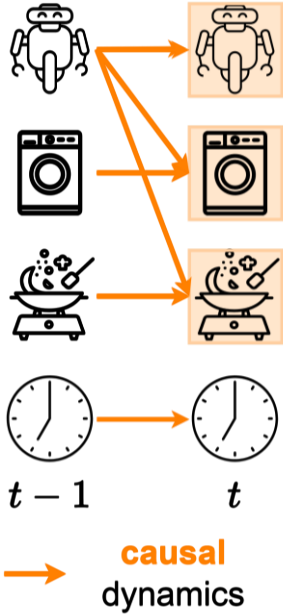
- learns a *causal* dynamics model $f: S_t \times A_t \rightarrow S_{t+1}$
- the state abstraction identifies and keeps all *controllable* state variables

minimal / sufficient ×

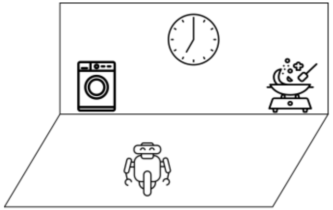
- includes an extra appliance for each task
- doesn't include clock

reusable ✓

- dynamics (and derived abstraction) are task-independent



Prior Work 2



TIA & Denoised MDPs

Fu et al, "Learning task informed abstractions" ICML 2021.
Wang et al, "Denoised MDPs: Learning world models better than the world itself" ICML 2022.

(TIA) During task learning:

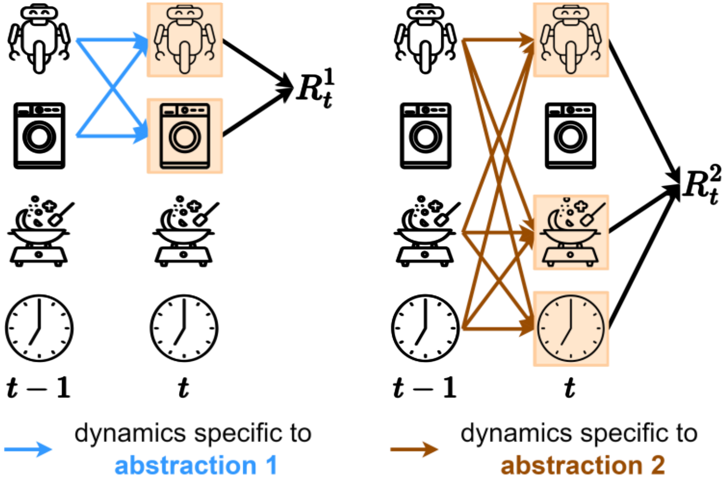
- Identify a minimal set of state variables that can predict rewards and the state variables' own dynamics

minimal/sufficient ✓

- by analyzing relevance to the reward

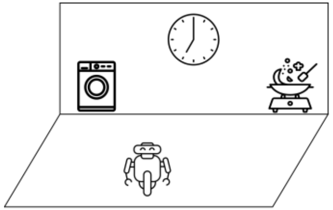
reusable ×

- dynamics models are specific to reward-relevant state variables



Zizhao Wang, **Caroline Wang**, Xuesu Xiao, Yuke Zhu, Peter Stone, AAAI 2024

Causal Bisimulation Modeling (CBM)



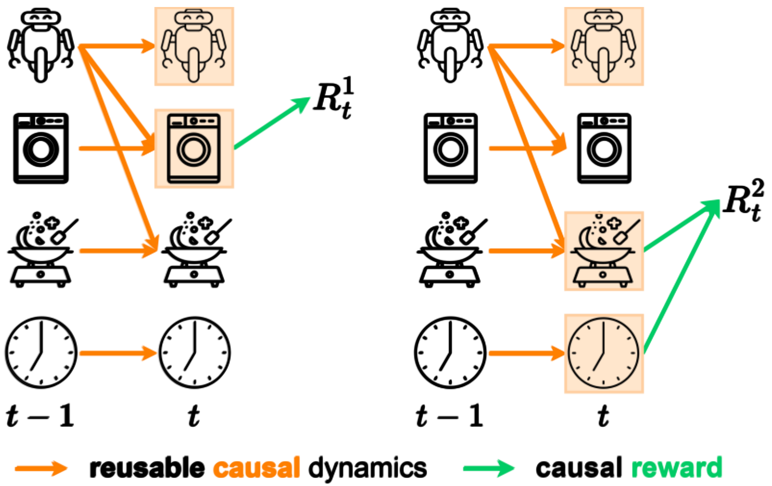
Can we combine the best of both worlds?

reusable ✓

– learn a causal dynamics model

minimal/sufficient ✓

– given a task, learn a causal reward model to identify reward relevant variables

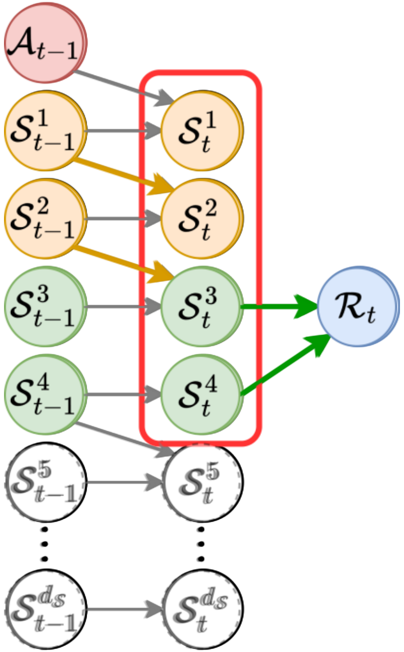


Zizhao Wang, **Caroline Wang**, Xuesu Xiao, Yuke Zhu, Peter Stone, AAAI 2024

Method (CBM) – causal state abstraction

Given causal dynamics and reward models, derive the **state abstraction** as all **ancestors** of the reward:

- parent variables affecting the reward
- ancestor variables affecting the parents via dynamics

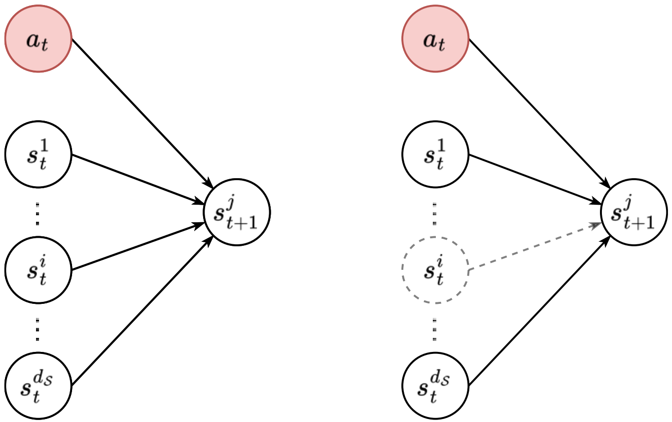


Method (CBM) – causal dynamics/reward models

Causality **all-but-one** test:

The dynamics/reward causal edge $s_t^i \rightarrow s_{t+1}^j$ or $s_t^i \rightarrow r_t^j$ exists
if s_t^i is necessary for prediction.

For example, to determine if a dynamics edge $s_t^i \rightarrow s_{t+1}^j$ exists,



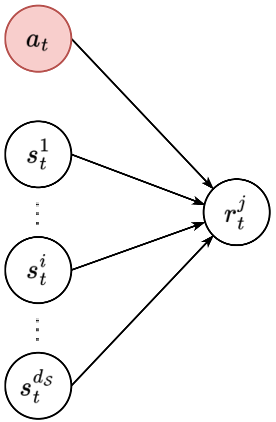
$$p\left(s_{t+1}^j \middle| s_t, a_t\right) \overset{?}{\approx} p\left(s_{t+1}^j \middle| \{s_t / s_t^i, a_t\}\right)$$

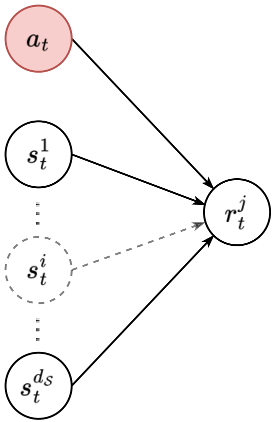
Method (CBM) – causal dynamics/reward models

Causality **all-but-one** test:

The dynamics/reward causal edge $s_t^i \rightarrow s_{t+1}^j$ or $s_t^i \rightarrow r_t^j$ exists if s_t^i is necessary for prediction.

Similarly, to determine if a reward edge $s_t^i \rightarrow r_t^j$ exists,





$$p\left(r_t^j \middle| s_t, a_t\right) \overset{?}{\approx} p\left(r_t^j \middle| \{s_t / s_t^i, a_t\}\right)$$

conditional mutual information (CMI)

$$\mathrm{CMI} = \mathbb{E}_{s,a,r}\left[\log \frac{p\left(r_t^j \middle| s_t, a_t\right)}{p\left(r_t^j \middle| \{s_t / s_t^i, a_t\}\right)}\right]$$

Method (CBM) – implicit dynamics model

implicit dynamics models $\hat{s}_{t+1} = \mathrm{argmax}_{s_{t+1}} g(s_{t+1}; s_t, a_t)$

where g is a scalar scoring function

vs

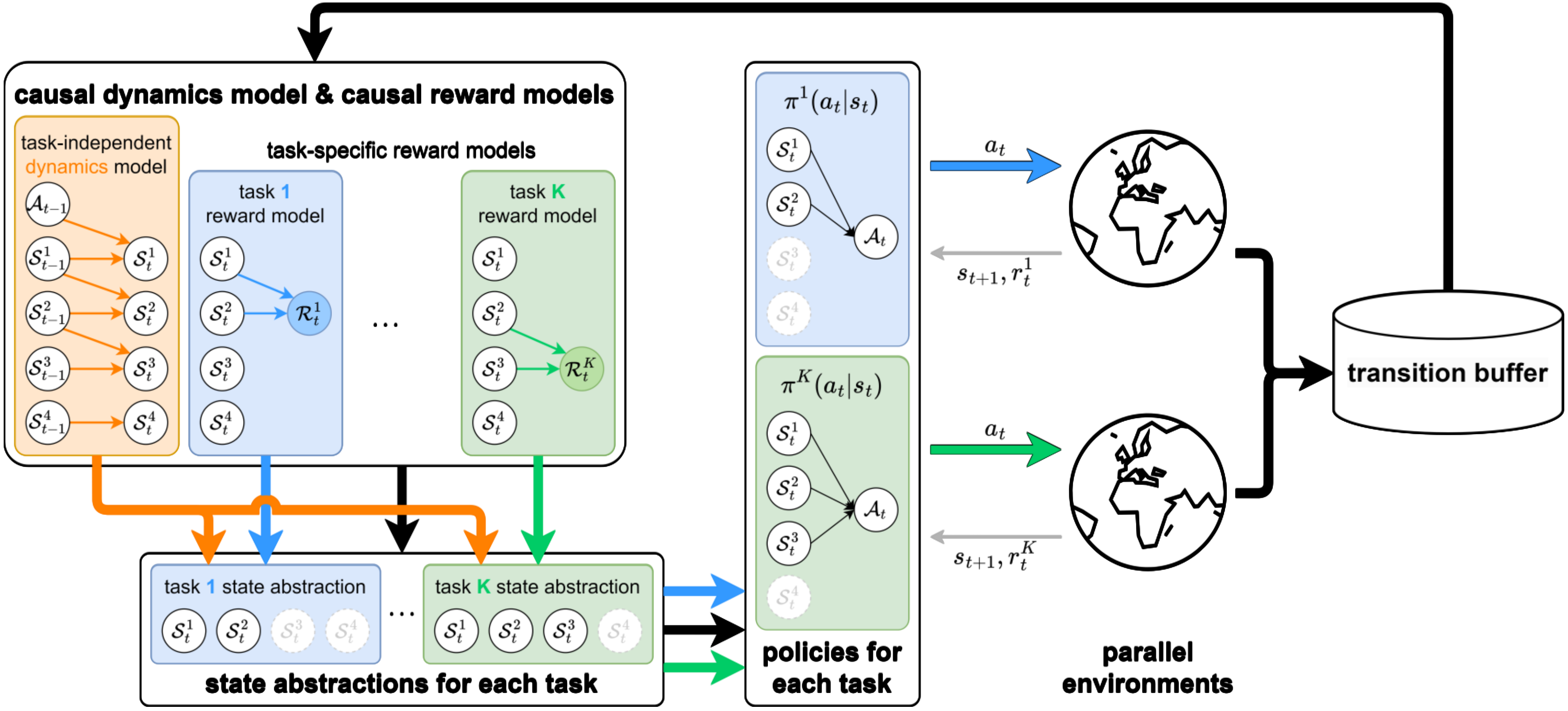
explicit dynamics models $\hat{s}_{t+1} = f(s_t, a_t)$

in a nutshell:

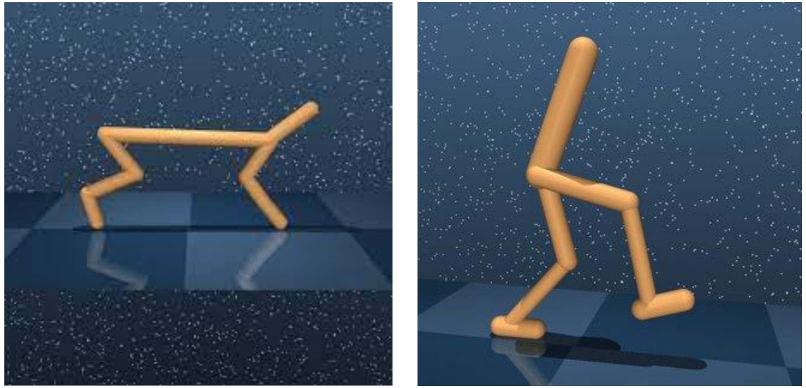
- introduce a method to model *implicit* causal dynamics
- find that implicit dynamics models are more accurate than explicit models

check our paper for details.

Method (CBM)

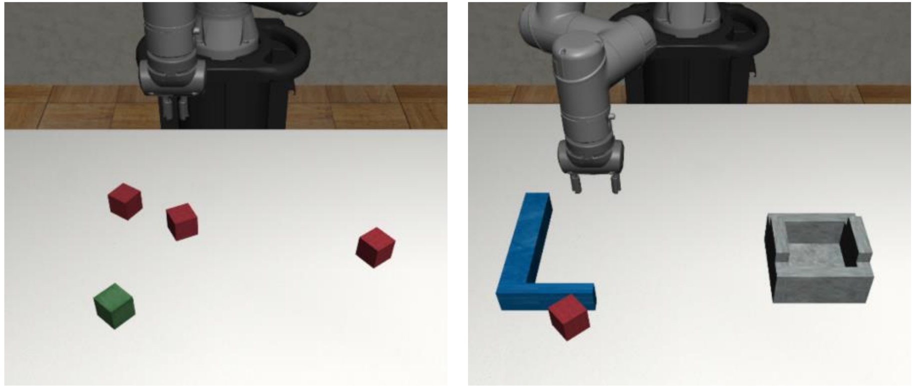

causal dynamics model & causal reward models
task-independent dynamics model
task-specific reward models
task 1 reward model
task K reward model
task 1 state abstraction
task K state abstraction
state abstractions for each task
$\pi^1(a_t|s_t)$
$\pi^K(a_t|s_t)$
policies for each task
a_t
s_{t+1}, r_t^1
s_{t+1}, r_t^K
parallel environments
transition buffer

Results



DeepMind control suite

Tassa et al. "Deepmind control suite.“ arXiv 2018.



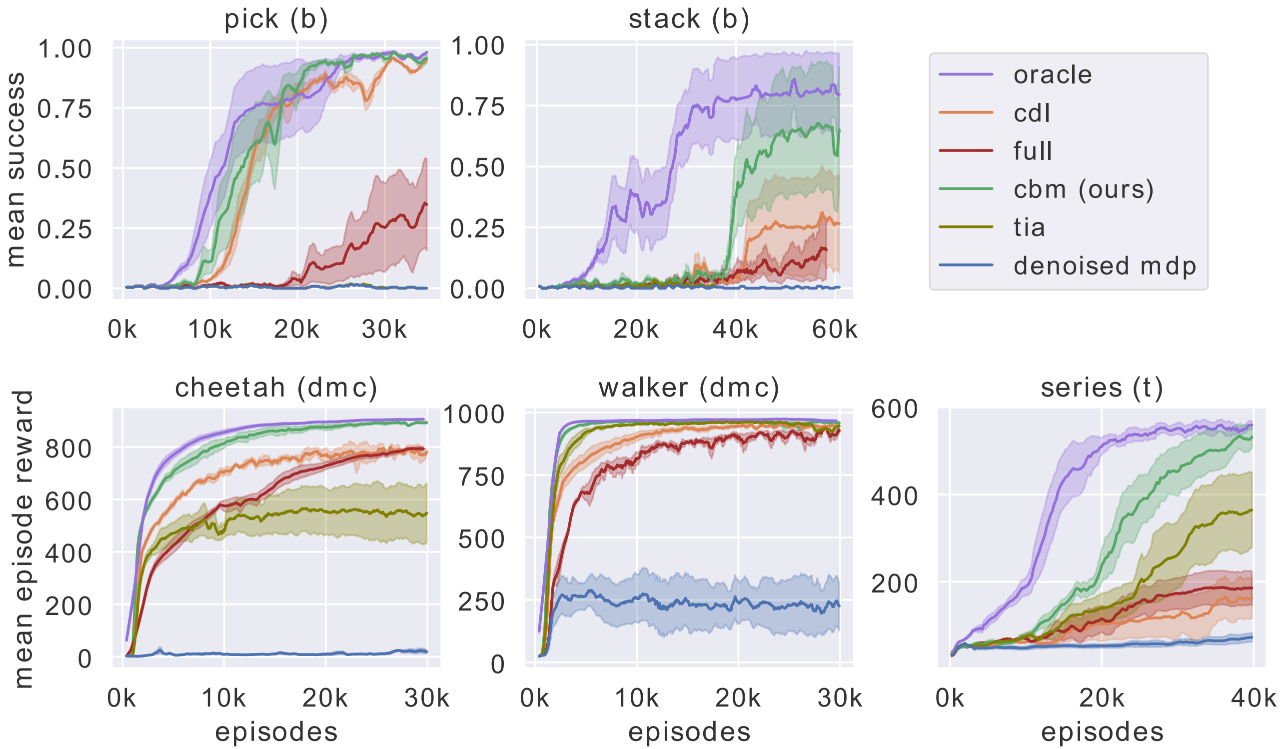
Robosuite table-top manipulation

Zhu et al “Robosuite: A modular simulation framework and benchmark for robot learning." arXiv 2020.

- Tasks: HalfCheetah, Walker
- Uncontrollable (20) and controllable noise variables (20)
- High-dimensional

- Environments: block (b), tool-use (t)
- Tasks: pick (b), stack (b), series (t)
- Pick/stack: moveable and unmovable blocks
- Series: long horizon

Results - task learning sample efficiency



Zizhao Wang, **Caroline Wang**, Xuesu Xiao, Yuke Zhu, Peter Stone, AAAI 2024

Thank you!

Building Minimal and Reusable Causal State Abstractions for Reinforcement Learning

(AAAI 2024)

Zizhao Wang*, **Caroline Wang***, Xuesu Xiao, Yuke Zhu, and Peter Stone





LARG Learning Agents Research Group The University of Texas at Austin | RPL | Sony AI

Results – effect of implicit vs explicit dynamics models on causality & abstraction accuracy

	block			tool-use	
	causal graph	pick	stack	causal graph	series
explicit	87.5 ± 0.1	53.2 ± 4.6	59.6 ± 4.6	82.6 ± 0.2	80.0 ± 1.5
implicit (ours)	**90.5** ± 0.4	**95.7** ± 6.0	**95.7** ± 6.0	**85.5** ± 0.1	**98.8** ± 1.3

Table 1: Mean ± std. error of accuracy (↑) for learned dynamics causal graphs and task abstractions.

causal graph accuracy = correctly classified graph edges / all possible edges
abstraction accuracy = correctly categorized state variables / all state variables