Resource-Rational Abstraction of Causal Models

Research Project Report

Archana Warrier
(Matriculation No.: 423618)

September 30, 2024

Contents

1 Introduction

The ability to create efficient mental models by filtering out irrelevant details is a fundamental aspect of human cognition. This ability allows us to reason effectively about complex systems without the need for perfect models of reality. Our research aims to bridge the gap between cognitive science and computational modeling by leveraging these human-like abstraction abilities to enhance simulation-based inference systems.

Drawing inspiration from Battaglia et al. [1], who proposed that human understanding of physical scenes relies on mental simulations, this work explores how to incorporate the efficiency of mental simulations into computational models. The focus is on resource-rational abstraction [2, 3] - a method for simplifying complex physical systems in a resource-rational manner wherein answers to queries on physical simulations are computed with predictive accuracy while maintaining low computational costs.

To illustrate the concept of rational abstraction, consider the experiments conducted by Battaglia et al. [1], where participants predicted the stability and potential direction of fall of block towers. Humans can intuitively answer these queries without simulating the precise physics of every block on the table. This aligns with findings by Shin and Gerstenberg [4] that suggest we employ abstraction rather than detailed physical simulation. We could simulate the block towers in two ways: using a highly detailed physics model that's computationally intensive, or employing a simplified model (such as representing towers as point masses) that's computationally more efficient but potentially less accurate for certain queries. The second model is more resource-rational and better represents human reasoning than the first.

This work explores resource-rational abstraction in two key ways:

1. Maximizing expected utility with respect to posterior belief over problems
2. Incorporating the cost of optimizing abstractions to ensure computational tractability

Through this research, we take a significant step towards building AI systems that incorporate human-like reasoning capabilities, enhancing both performance and interpretability in complex problem-solving tasks.

2 Background

Two key but related notions of abstraction has been developed independently, one in programming languages and another in causality literature. We explore both in a brief literature review in this section.

2.1 Abstraction for Model Checking

Clarke [5] provides a definition for abstractions of transition systems as a means to verify correctness properties of the execution of programs. Specifically, the aim is to construct an abstract model of a program, in the form of a transition system, and verify the correctness properties against the model as a proxy for the original program. In their formalism, a transition system is modeled as a tuple (S, I, R) where S is a set of states, $I \subseteq S$ is a set of initial states, and $R \subseteq S \times S$ is a transition relation. They further decompose S into a set of program variables. That is, $S = D = D_1 \times D_2 \times \cdots \times D_n$ where D_i denotes the set of possible values of the ith program variable.

To abstract a transition system, they first define sets $\hat{D}_i$ of abstract values, and a surjection $h_i : D \to \hat{D}$. An abstract transition model $\hat{M} = (\hat{S}, \hat{I}, \hat{R})$ is an abstraction of M if the following conditions hold:

1. $\exists d.\hat{d} = h(d) \wedge I(D) \implies \hat{I}(\hat{d})$
2. $\exists d_1, d_2.h(d_1) = \hat{d}_1, h(d_2) = \hat{d}_2, R(d_1, d_2) \implies \hat{R}(\hat{d}_1, \hat{d}_2)$

In words: $\hat{M}$ is a valid abstraction if starting at some arbitrary concrete state d, two routes result in the same abstract state. The first route takes a concrete transition according to the transition relation R and then abstracts the post-transition state. The second route abstracts d and then takes an abstract transition. The intuition is first that an abstract state corresponds to a *set* of concrete states. Then, an abstract transition between these abstract states is valid if it over-approximates.

2.2 Causal Abstraction

More recently, formal notions of abstraction have been explored within the context of structural causal models. Some background on the definitions of causal models is given below:

Definition 2.1 (Signature)**.** *A* signature $\mathcal{S}$ *is a tuple* $(\mathcal{U}, \mathcal{V}, \mathcal{R})$ *where* $\mathcal{U}$ *is a set of exogenous variables,* $\mathcal{V}$ *is a set of endogenous variables and* $\mathcal{R}$ *is a function such that maps every variable in* $\mathcal{U} \cup \mathcal{V}$ *with a non-empty set of values it could take.*

Definition 2.2 (Causal Model)**.** *A* causal model $\mathcal{M}$ *is defined as* $(\mathcal{S}, \mathcal{F}, \mathcal{I})$ *where* $\mathcal{S}$ *is a signature,* $\mathcal{F}$ *denotes the structural equations and* $\mathcal{I}$ *is the set of interventions that are possible in the real-world or interventions that we care about.*

Formally, the equation $\mathcal{F}_X$ maps $\mathcal{R}(\mathcal{U} \cup \mathcal{V} - X)$ to $\mathcal{R}(X)$, so $\mathcal{F}_X$ determines the value of X, given the values of all the other variables in $\mathcal{U} \cup \mathcal{V}$. Note that there are no functions associated with exogenous variables; their values are determined outside the model. We call a setting $\vec{u}$ of values of exogenous variables a *context.*

Definition 2.3 (Probabilistic Causal Model)**.** *A* probabilistic causal model *is a tuple* $(\mathcal{M}, Pr)$ *where* $\mathcal{M}$ *is a causal model and* Pr *is a probability on the context.*

Beckers and Halpern [6] provide definitions for causal abstraction and constructive causal abstractions. The intuition there is that every intervention $i \in \mathcal{I}_L$ acts the same way in the low-level model as the intervention $\omega(i)$ does in the high level-model. i.e., For all interventions in the low level model, if you start at that intervention and move to the high level model following two routes, you end up at the same place. The first route applies τ (a surjective pushforward to ensure the high-level model has fewer variables that the low-level model) to the interventional distribution on low level outcomes, giving $\tau(Pr_L^i)$, which is a distribution on high level outcomes. The second route transforms the low level intervention into a high level one by applying ω, which changes the probability distribution on high level outcomes, giving $Pr_H^{\omega(i)}$.

A high-level causal model, M_H is a *constructive* τ-abstraction of a low-level causal model, M_L if M_H is a τ-*abstraction* of M_L and $\mathcal{V}_H$ is a partition set of $\mathcal{V}_L$. Beckers and Halpern [6] further say that an abstract causal model can be constructed from a low-level model by performing the following operations compositionally:

1. Marginalization: Ignoring low-level variables,
2. Variable merge: Combining multiple low-level variables into a single high-level variable, and
3. Value merge: Collapsing multiple values in a variable's domain into a single value.

3 Problem Overview

Given the two different notions of abstraction, we want the generality and theoretical background behind of abstracting causal models but be able to use it in practice like abstraction in model checking. In practice, humans use the causal model that has the necessary information to answer some query correctly [4], which we call the *resource-rational abstraction*, derived from the definitions of resource-rationality [2, 3] as exhibited by humans.

In this project, we consider the case where we have the low-level model. We still want to be able to traverse its abstraction ladder, viewing a coarser or finer version of the model depending on how we wish to use it. Another way we restrict the problem so that it is well-defined is by considering only physics simulation environments (MuJoCo [7]) as our causal models. The domain is complicated enough to be a good use-case of resource-rational abstraction and has well-defined, meaningful abstractions similar to the ones in model checking papers.

4 Formalized Problem Statement

4.1 General Definitions

Definition 4.1 (Model Space)**.** *Let $\mathcal{M}$ denote a space of models. A model, $m \in \mathcal{M}$ is a causal model that describes the scene that is to be simulated.*

Definition 4.2 (Query Space)**.** *Let $\mathcal{Q}$ denote a space of queries. A query $q \in \mathcal{Q}$ is a function that takes a model m and maps it to the a value of type $\mathcal{V}$.*

Definition 4.3 (Problem Space)**.** *Let $\mathcal{P} = \mathcal{M} \times \mathcal{Q}$ denote the space of problems.*

Definition 4.4 (Solver)**.** *A solver is a function $s : \mathcal{P} \to \mathcal{V}$ that produces a solution to a problem, ie., computes the query on the model.*

4.2 Abstraction and Utility Functions

Definition 4.5 (Abstraction Function)**.** *An abstraction function maps a problem to an abstract problem , i.e., it is a function, $\alpha : \mathcal{P} \to \mathcal{P}$, where $\alpha(m, q) = (\hat{m}, \hat{q})$.*

Definition 4.6 (Approximation Utility)**.** *The utility of an abstraction relative to a problem (m, q) is:*

$$U_{approx}(m, q, \hat{m}, \hat{q}) = \ell(s(m, q), s(\hat{m}, \hat{q}))$$

where ℓ is a function comparing the accuracy of evaluation of the high-fidelity problem and it's abstraction.

Definition 4.7 (Resource-Rational Utility)**.** *Balances approximation error and resources:*

$$U(m, q, \hat{m}, \hat{q}) = U_{approx}(m, q, \hat{m}, \hat{q}) - \lambda C(\hat{m}, \hat{q})$$

where $C(\hat{m}, \hat{q})$ is the cost of using the abstraction and λ is a real-valued parameter.

Definition 4.8 (Computational Cost)**.** *For concreteness, we consider our costs to be computational costs. Given some expression e, let $[\![e]\!]_{Cost}$ denote the cost of evaluating e. Then:*

$$C(\hat{m}, \hat{q}) = [\![s(\hat{m}, \hat{q})]\!]_{Cost}$$

4.3 Optimal Abstraction

Definition 4.9 (Resource-Optimal Abstraction)**.** *The resource-optimal abstraction of a problem, (m, q) is:*

$$\alpha^*(m, q) = \arg\max_{\hat{m}, \hat{q}} U(m, q, \hat{m}, \hat{q})$$

Given a sequence of problems $p_1, p_2, \ldots, p_n$, we can define the utility of the sequence for a given abstraction function α as:

$$U(\vec{p}, \alpha) = \sum_i U(p_i, \alpha(p_i))$$

where $p_i = (m_i, q_i)$ *and* $\alpha(p_i) = (\hat{m}_i, \hat{q}_i)$

Therefore, the resource optimal abstraction of a problem sequence is:

$$\alpha^*_{seq} = \arg\max_{\alpha} \frac{1}{n} \sum_{i=1}^{n} U(p_i, \alpha(p_i))$$

where α^*_{seq} *is the abstraction function that maximizes the average utility across the entire sequence of problems.*

4.4 Meta-Optimization

To account for the computational costs of evaluating the abstraction function itself, we introduce meta-optimization:

Definition 4.10 (Abstraction Evaluation Cost)**.** *Let* $C_\alpha(p)$ *denote the cost of evaluating the abstraction function* α *on problem* p*.*

Definition 4.11 (Meta-Resource-Rational Utility)**.** *We define a new utility function that incorporates the cost of abstraction itself:*

$$U_{meta}(p, \alpha) = U(p, \alpha(p)) - \mu C_\alpha(p)$$

where μ *is a meta-level trade-off parameter.*

Definition 4.12 (Meta-Resource Optimal Abstraction)**.** *For a sequence of problems* $\vec{p} = (p_1, \ldots, p_n)$*, the meta-resource optimal abstraction is:*

$$\alpha^*_{meta} = \arg\max_{\alpha} \frac{1}{n} \sum_{i=1}^{n} U_{meta}(p_i, \alpha)$$

This can be expanded as:

$$\alpha^*_{seq} = \arg\max_{\alpha} \frac{1}{n} \sum_{i=1}^{n} [\ell(s(p_i), s(\alpha(p_i))) - \lambda C(s(\alpha(p_i))) - \mu C_\alpha(p_i)]$$

5 Experiments

As a proof of concept, we conducted an experiment in MuJoCo [7] wherein we have a scene with a discrete set of abstractions. The problem then becomes a combinatorial one of choosing the right abstraction given a query. In the example, the scene has a static table and two hollow cups falling from a certain (probabilistic) height as given in 1. The masses of the cups are also random variables. The high-level model uses fewer boxes to approximate the cups' geometries, which is similar to a variable merge operation in the causal abstraction literature.

The cost functions in this experiment are simplified functions of the number of boxes used to approximate the object, which is a coarse approximation. This is not

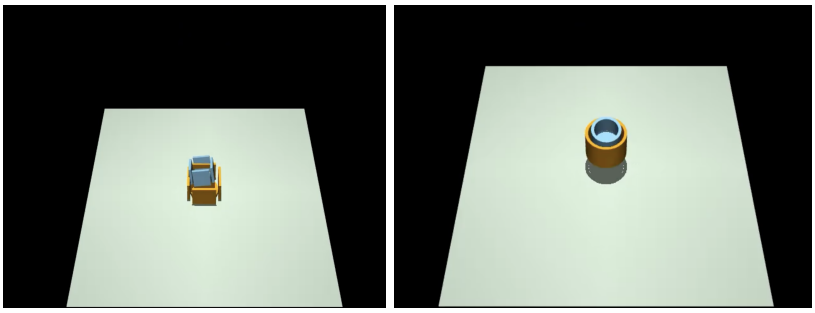

Figure 1: The figure shows the two extreme models or scenes in the abstraction spectrum. The highest level of abstraction uses four boxes to approximate the cups while the low-level model uses 30 boxes.

generally true, and in a more general abstraction setting has been implemented as a function of the entire scene itself (which includes velocities of objects, positions of objects, collisions, etc). This experiment simply gives an intuition for why we should care about resource-rational abstractions. Even in an over-simplified setting such as this one, we can clearly see the "right" abstraction we should choose should the queries remain the same as those in 2. From 2 we see that if we care about answering all three queries relatively accurately while being cost-aware, the abstraction we should choose approximately uses 16-17 boxes to form the cups.

6 Discussion and Future Work

This project introduces a novel approach to accelerating physics simulations through resource-rational abstraction, inspired by human cognitive processes. Though the results are preliminary, they demonstrate improvements in computational efficiency while maintaining a high accuracy while solving inference problems based on the simulation. In practice for the above experiment it's in the order of a few minutes for about 50 samples of the scene, which is significant considering how simple the scene is. Ongoing work is being done on using gradient based methods to find the abstraction function α, instead of an abstract model, which is more general as it can be applied to new scenes that we haven't seen before.

Future work beyond that includes:

- developing more general abstractions that could be applied at a higher level than geometry, incorporating simulation parameters (e.g., solver or timestep used during the simulation),
- expanding the space of allowed queries by finding meaningful query abstractions and including interventional queries, and
- exploring applications in planning and decision making.

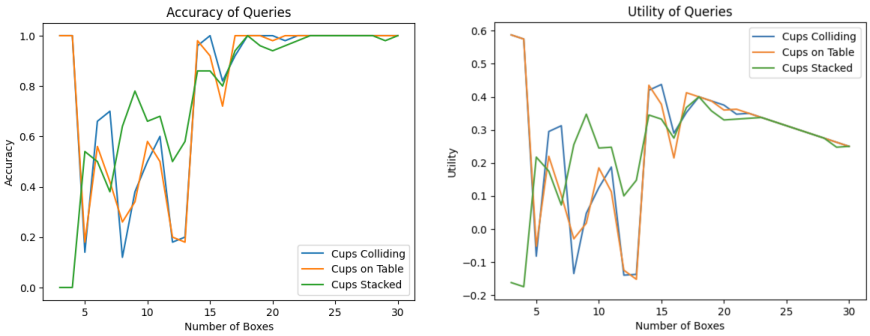

Figure 2: The figures shows the number of boxes used to approximate the cups v/s how accurately it answers the queries asked and the utility of each level of abstraction: (i) "Do the cups collide with each other?" (ii) "Are the cups on the table once the simulation stabilizes?", and (iii) "Do the cups stack perfectly one inside the other when the simulation stabilizes?"

References

[1] Peter W Battaglia, Jessica B Hamrick, and Joshua B Tenenbaum. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, 110(45):18327–18332, 2013.

[2] Thomas Icard. Resource rationality. *Book manuscript*, 2023.

[3] Falk Lieder and Thomas L Griffiths. Resource-rational analysis: Understanding human cognition as the optimal use of limited computational resources. *Behavioral and brain sciences*, 43:e1, 2020.

[4] Steven M Shin and Tobias Gerstenberg. Learning what matters: Causal abstraction in human inference. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 45, 2023.

[5] Edmund M Clarke. Model checking. In *Foundations of Software Technology and Theoretical Computer Science: 17th Conference Kharagpur, India, December 18–20, 1997 Proceedings 17*, pages 54–56. Springer, 1997.

[6] Sander Beckers and Joseph Y. Halpern. Abstracting causal models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):2678–2685, 7 2019. doi: 10.1609/aaai.v33i01.33012678. URL `https://ojs.aaai.org/index.php/AAAI/article/view/4117`.

[7] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.